

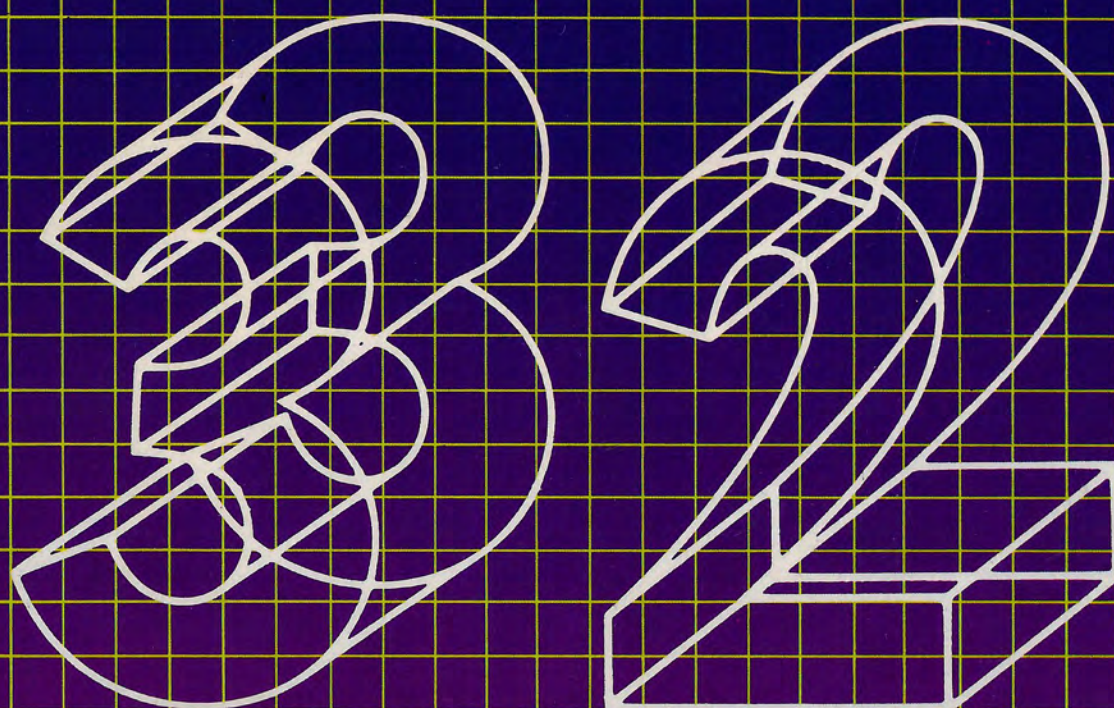
BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,

IBM, SPECTRUM, COMMODORE Y MSX



DIVERTIDOS PROGRAMAS

DE BOTANICA, LITERATURA, FISICA, ETC.

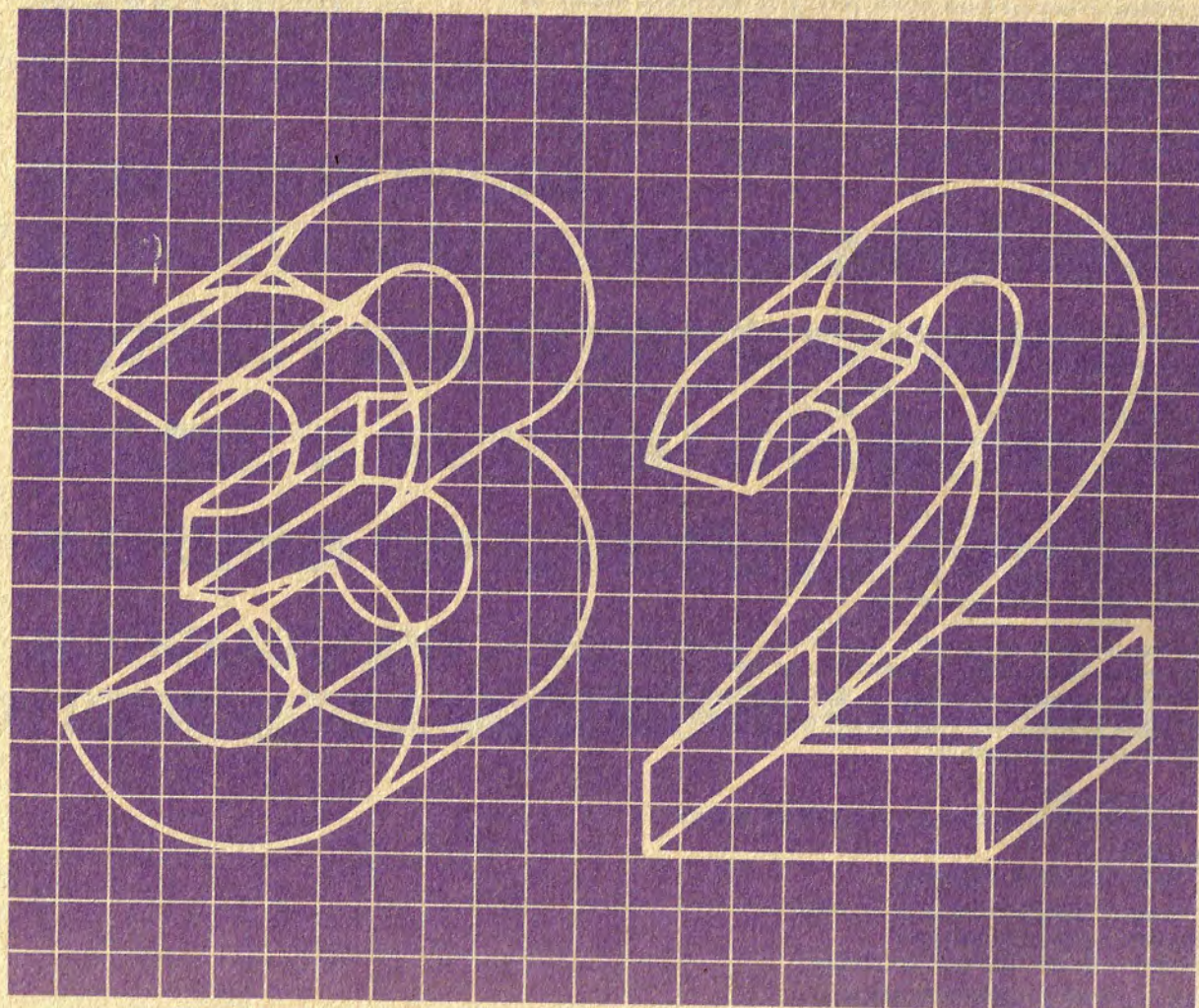
EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 32. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubelo, Técnica de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: Jesús Bocho Licenciado en Informática; Alejandro Marcos, Licenciado en Química. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 455 09 99. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-171

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

GRAFICAS EMA. (Madrid)

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 259 73 31. 28020 Madrid.

Noviembre, 1987

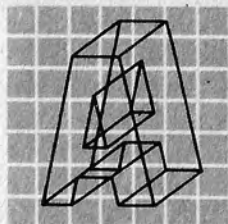
P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	14
■ TRUCOS Y RUTINAS BASICAS	23
■ EL TALLER DEL HARDWARE	36
■ APRENDER CON EL ORDENADOR	42
■ VOCABULARIO DE INFORMATICA	53

EXPERIENCIA Y PRACTICAS EN LOGO

■ Listas de propiedades



NTES de continuar con las listas de propiedades, recordemos algunos conceptos ya vistos en el tomo anterior.

— En LOGO a las variables y a los procedimientos podemos asociarles una

lista de propiedades.

— La estructura es la siguiente:

[propiedad 1 valor 1 propiedad 2 valor 2...]

Las listas de propiedades contienen pares de elementos formados por una propiedad y su valor.

— No es posible asociar una lista de propiedades a una variable que no tenga ningún valor.

— Si borramos del espacio de memoria una variable que tiene una lista de propiedades asociadas, las perdemos.

— PONPROP nombre propiedad objeto.

— Esta primitiva permite asociarle a la variable o al procedimiento especificado en "nombre, una propiedad con un valor "objeto determinado.

— PROP nombre propiedad.

Esta primitiva devuelve el valor de la propiedad especificada de la lista asociada al nombre indicado.

NOTA: El LOGO del SPECTRUM no dispone de la posibilidad de asociar listas de propiedades a las variables y/o a los procedimientos.

■ Borrar listas de propiedades

Una primitiva que nos permite borrar propiedades es:

BOPROP nombre propiedad.

significa BORra PROPIedad.

Con esta primitiva borramos de "nombre la propiedad especificada y el valor asociado a ella.

Veamos algunos ejemplos:

Primero asociamos a una variable una lista de propiedades.

?HAZ "A CASA

?PONPROP :A "BALCON "EXTERIOR

?PONPROP :A "TELEFONO "ROJO

Comprobamos que la variable A tiene asociadas estas propiedades con sus respectivos valores.

?ES PROP :A "TELEFONO

ROJO

ROJO es el valor de la propiedad TELEFONO.

?ES PROP :A "BALCON

EXTERIOR

Borra de una lista de propiedades las que no te interesen.

EXPERIENCIA Y PRACTICAS EN LOGO

EXTERIOR es el valor de la propiedad BALCON.

Ahora borramos de la variable A la propiedad BALCON.

```
? BOPROP :A "BALCON
```

Comprobamos que la variable A ya no tiene esta propiedad.

```
? ES PROP :A "BALCON
```

En la pantalla no aparece nada, es decir, la variable A ya no tiene la propiedad BALCON.

También podemos borrar la otra propiedad que tiene asociada la variable A.

```
?BOPROP :A "TELEFONO
```

Si intentamos ver el valor de la propiedad "TELEFONO, en la pantalla no aparecerá nada.

```
?ES PROP :A "TELEFONO
```

Hemos borrado por separado las propiedades que anteriormente habíamos asociado a la variable A.

Si lo que queremos es borrar la lista entera de propiedades de una variable o de un procedimiento, podemos utilizar otra primitiva que nos ofrece el LOGO, ésta es:

```
BOPROPS nombre
```

que significa BORra PROPIedades

Borra todas las propiedades del nombre indicado.

Si volvemos a asociar a la variable A las propiedades anteriores, podemos comprobarlo.

```
?PONPROP :A "BALCON "EXTERIOR
```

```
.. ?PONPROP :A "TELEFONO "ROJO
```

Ahora ya podemos borrarlas.

```
?BOPROPS :A
```

Veamos otro ejemplo:

```
? PARA "A1
```

```
> HAZ "A [ME ENCANTAN LAS ZANAHORIAS]
```

```
> ES :A
```

```
> FIN
```

Ahora le asignamos el procedimiento A1 una serie de propiedades.

```
? PARA A2
```

```
> PONPROP "A1 "PADRE [JOSE ANTONIO]
```

```
> PONPROP "A1 "MADRE [MARIA LUISA]
```

```
> ES PROP "A1 "PADRE
```

```
> ES PROP "A1 "MADRE
```

```
> FIN
```

Si ejecutamos este procedimiento:

```
? A2
```

En la pantalla aparece:

JOSE ANTONIO

MARIA LUISA

```
? PARA A3
```

```
> BOPROP "A1 "PADRE
```

```
> BOPROP "A1 "MADRE
```

```
> ES PROP "A1 "PADRE
```

```
> ES PROP "A1 "MADRE
```

```
> FIN
```

Y cuando ejecutamos A3 en la pantalla no aparecerá nada.

```
? A3
```

Lo mismo si hubiésemos conseguido con este otro procedimiento:

```
? PARA A3
```

```
> BOPROPS "A1
```

```
> ES PROP "A1 "PADRE
```

```
> ES PROP "A1 "MADRE
```

```
> FIN
```

■ Particularidades de la versión de LOGO para el Spectrum

La versión de Logo para el SPECTRUM presenta una serie de primitivas muy específicas y que sólo son válidas para este ordenador.

Como ya conocerán los usuarios de este ordenador, la pantalla se encuentra dividida en dos zonas, el borde y el papel. A cada una

Si borramos una propiedad de la lista asociada a una variable o a un procedimiento, también borramos su valor.

de estas zonas podemos asignarles colores diferentes.

Las órdenes que se utilizan en Basic para asignar estos colores son:

- BORDER *n*: Para el borde
- PAPER *n*: Para el papel

Las correspondientes primitivas que utiliza el Logo para realizar estas acciones son:

- PONCOLORBORDE *n*: Para el borde
- PONCOLORFONDO *n*: Para el papel

Veamos detenidamente cada una de estas primitivas.

PONCOLORBORDE se puede escribir de la forma abreviada:

PONCB (PON Color Borde)

Esta orden necesita un parámetro (*n*) para que sea ejecutable. *n* determinará el color que se desea que tenga el borde de la pantalla y debe coincidir con uno de los códigos de colores que tiene este ordenador.

Recordemos los códigos de colores de que disponía:

- 0— Negro
- 1— Azul
- 2— Rojo
- 3— Magenta
- 4— Verde
- 5— Cyan
- 6— Amarillo
- 7— Blanco

PONCOLORFONDO se puede escribir también de forma abreviada:

PONCF (PON Color Fondo)

Al igual que la primitiva anterior, ésta también precisa un parámetro (*n*) para que se pueda ejecutar, y debe corresponder a uno de los códigos de colores que hemos visto.

Otra particularidad que presenta este ordenador es que podemos escribir los textos que van a aparecer en la pantalla con mayor intensidad de la normal, es decir, con brillo.

La orden que el Basic nos permite esto es:

- BRIGHT *n*

La correspondiente primitiva en Logo es:

? PONBRILLO *n*

El valor del parámetro *n* que utiliza esta orden sólo puede presentar dos valores, 0 ó 1.

POMBRILLO 1 activará el brillo, y todo lo que se escriba a partir de este momento se hará en papel brillante.

Para dejar de escribir de esta forma, tenemos que desactivar el brillo. Esto lo conseguimos al ejecutar:

? PONBRILLO 0

Otra posibilidad que nos ofrece este ordenador es la de escribir los textos en forma centelleante, en flash.

En Basic podemos obtener este efecto con la orden:

— FLASH *n*

en donde *n* puede tener el valor 1 ó 0. Si se especifica un 1, indicamos que se active este efecto, y si se especifica un 0, que se desactive.

En el Logo esta orden se escribe de la misma forma, pero no precisa de un parámetro para que se pueda ejecutar.

Si introducimos la primitiva:

? FLASH

todo lo que se escriba a partir de este momento se hará en forma centelleante.

Para desactivar este efecto tenemos que utilizar otra primitiva. Se trata de:

? NORMAL

Con esta primitiva conseguimos que se desactive el flash y que todo lo que a partir de este momento se escriba se haga de forma normal.

El efecto de flash también se desactivará cuando se termine la ejecución del procedimiento en el que se encuentra o se produzca el error.

Por último, este ordenador nos permite escribir los textos invirtiendo los colores del fondo del texto y del texto.

En Basic esto lo conseguimos con la orden:

— INVERSE *n*

Al igual que ocurría con FLASH, el parámetro *n* sólo podrá tener dos valores, 0 ó 1.

Es posible borrar de golpe toda una lista de propiedades.

EXPERIENCIA Y PRACTICAS EN LOGO

Si se especifica un 1, se activará la forma inversa, y si se especifica un 0, se desactivará.

En el Logo la primitiva que realiza esta acción es:

? INVERSO

Este caso tampoco precisa de un parámetro para que se ejecute.

Al introducir esta primitiva se activará el modo de escritura inverso y todo lo que se escriba a partir de este momento aparecerá con los colores intercambiados.

Para desactivar este efecto debemos introducir también la primitiva:

? NORMAL

Al igual que ocurrió con el flash, este efecto también desaparecerá cuando se termine de ejecutar el procedimiento en donde se encuentra o se produzca un error.

■ Cuadro resumen

— BOPROP nombre propiedad

Borra la propiedad con su valor del nombre especificado.

— BOPROPS nombre

Borra todas las propiedades del nombre especificado.

— PONCOLORBORDE *n* (PONCB *n*)

Sólo en la versión del Logo del SPECTRUM.

Pone del color que determina *n* el borde de la pantalla del ordenador.

— PONCOLORFONDO *n* (PONCF *n*)

Sólo en la versión del Logo del SPECTRUM

Pone del color que determina *n* la zona designada como papel de la pantalla.

— PONBRILLO *n*

Sólo en la versión del Logo del SPECTRUM.

Si el valor de *n* es 1, se activa el efecto brillo, y todo lo que se escriba aparecerá sobre papel brillante.

Si el valor de *n* es 0, se desactiva el efecto de brillo.

— FLASH

Sólo en la versión del Logo del SPECTRUM.

Activa el efecto de parpadeo y todo lo que se escriba se hará de forma centelleante.

— INVERSO

Sólo en la versión del Logo del SPECTRUM.

Activa el efecto de inversión del color del fondo con el color del texto.

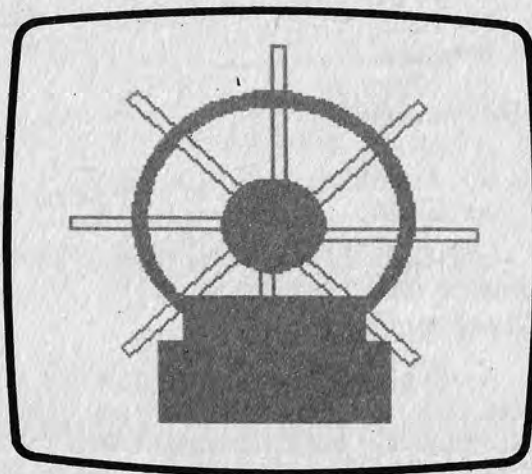
— NORMAL

Sólo en la versión del Logo del SPECTRUM.

Desactiva el efecto de parpadeo (FLASH) y de inversión de colores fondo-texto (INVERSO).

■ Ejercicios

1. Intenta dibujar una cometa.
2. Realiza el siguiente dibujo:



3. Define los procedimientos para realizar el siguiente proceso:
 - Se introducirá una palabra de cuatro letras.
 - Una vez introducida y tras pulsar una tecla, aparecerá en otra pantalla la palabra introducida.
 - Tras pulsar de nuevo una tecla, los ele-

Asociar y desasociar propiedades es fácil con el Logo.

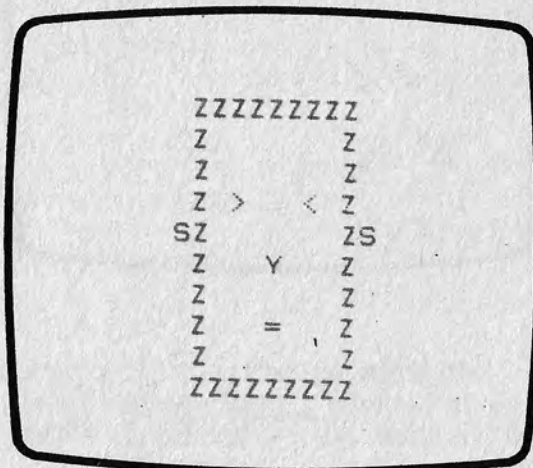
mentos de la lista se escribirán en el centro de la pantalla de la siguiente forma:

```

      1.º
4.º   2.º
      3.º
  
```

Tras una pequeña espera, los elementos deben empezar a rotar en el sentido de las agujas del reloj.

4. Define los procedimientos necesarios para dibujar:



5. ¿Son correctas las siguientes órdenes?:

- ? PARA A1
 - > HAZ "A PONPROP "A1 "SALUDOS "HOLA
 - > ES :A
 - > FIN
- ? PARA A2
 - > PONPROP "A2 "SALUDOS "ADIOS
 - > ES PROP :A2 "SALUDOS
 - > FIN
- ? PARA A3
 - > PONPROP "A3 "SALUDOS "ADIOS
 - > BOPROP "A3 "SALUDOS
 - > FIN
- ? PARA EFECTOS
 - > PONBRILLO 1
 - > FLASH
 - > INVERSO
 - > ESC "EFECTOS
 - > FIN

- ? FLASH 1
- ? PONCB 2
- ? PONCB 3
- NORMAL

■ Solución a los ejercicios

1:

El dibujo de la cometa lo realizamos mediante diversos procedimientos; todos ellos, y en un orden determinado, son ejecutados desde el procedimiento DIBUJO.

- ? PARA DIBUJO
 - > BP
 - > CRUZ
 - > PINTA
 - > CUERDA
 - > FIN

El procedimiento CRUZ dibuja la cruz central de la cometa y une los extremos entre sí.

- ? PARA CRUZ
 - > SL
 - > PONPOS [-16 52]
 - > PONRUMBO 90
 - > BL
 - > PONCL 1
 - > AV 14 GI 90
 - > AV 20 GD 90
 - > SL
 - > AV 4 GD 90
 - > BL
 - > AV 20 GI 90
 - > AV 14 GI 90
 - > SL
 - > AV 4 GD 90
 - > BL
 - > AV 14 GI 90
 - > AV 30 GD 90
 - > SL
 - > AV 4 GD 90
 - > BL
 - > AV 30 GI 90
 - > AV 14 GD 90
 - > SL
 - > PONPOS [-19 50]
 - > BL
 - > PONPOS [0 75]

Podemos asociarle propiedades a un procedimiento que no esté definido.

EXPERIENCIA Y PRACTICAS EN LOGO

```
> PONPOS [19 50]
> PONPOS [0 10]
> PONPOS [-19 50]
> SL
> PONPOS [0 50]
> BL
> RELLENA
> FIN
```

PINTA rellena de color las cuatro partes de la cometa.

El procedimiento se ha desarrollado en un PC-COMPATIBLE, si tú posees un ordenador que disponga de más colores, utilízalos.

```
? PARA PINTA
> SL
> PONPOS [-5 54]
> PONCL 2
> BL
> RELLENA
> SL
> PONPOS [5 54]
> BL
> PONCL 3
> RELLENA
> SL
> PONPOS [5 46]
> BL
> PONCL 3
> RELLENA
> SL
> PONPOS[5 46]
> BL
> PONCL 2
> RELLENA
> FIN
```

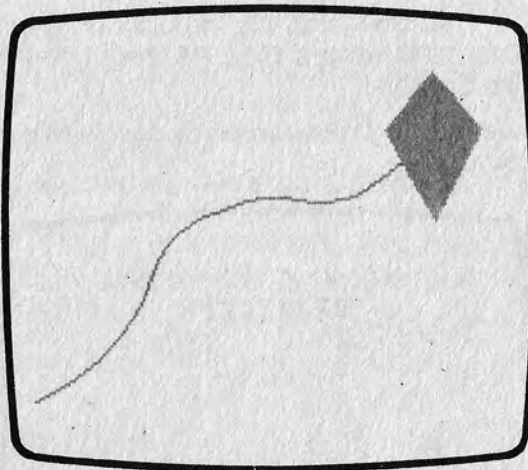
Y, por último, el procedimiento CUERDA que dibuja la cuerda de la cometa.

```
? PARA CUERDA
> SL
> PONPOS [2 48]
> PONRUMBO 230
> PONCL 2
> BL
> AV 30
> REPITE 5 [AV 6 GD 10]
> AV 5
> REPITE 3 [AV 7 GI 10]
> AV 10
> REPITE 5 [AV 7 GI 10]
```

```
> AV 5
> REPITE 5 [AV 15 GD 10]
> FIN
```

Para ver el dibujo tecleamos:

? DIBUJO



2:

Definimos un procedimiento para inicializar la pantalla y para ejecutar todos los procedimientos que van realizando el dibujo.

? PARA DIBUJO

```
> BP
> CAJA
> RUEDA
> PALOS
> FIN
```

El procedimiento CAJA dibuja la caja sobre la que se apoya el timón.

```
? PARA CAJA
> SL PONCL 3
> PONPOS [-35 -30]
> PONRUMBO 90
> BL
> REPITE 2 [AV 70 GD 90 AV 20 GD 90]
> SL
> PONPOS [-45 -50]
> BL
> REPITE 2 [AV 90 GD 90 AV 35 GD 90]
> SL
> PONPOS [-20 -35]
> BL
> RELLENA
```

En la versión del Logo del SPECTRUM podemos escribir con brillo en el texto.

```

> SL
> PONPOS [-30 -55]
> BL
> RELLENA
> FIN

```

RUEDA dibuja las circunferencias necesarias para dibujar el timón.

```

? PARA RUEDA
> SL
> CENTRO
> BL
> PONCL 3
> HAZ "R 20
> CIR :R
> BL
> PONCL 2
> RELLENA
> PONCL 3
> HAZ "R 50
> CIR :R
> HAZ "R 15
> CIR :R
> HAZ :R 5
> CIR :R
> HAZ :R 55
> CIR :R
> PONCL 3
> AV 55
> BL
> RELLENA
> FIN

```

Para dibujar los círculos, en el procedimiento RUEDA recurrimos al procedimiento CIR.

Este procedimiento dibuja las circunferencias mediante el radio, que es el dato de entrada del procedimiento.

```

? PARA CIR :R
> SL
> PONRUMBO 270
> AV :R
> PONRUMBO 0
> BL
> HAZ "L 6.20 *: R / 36
> REPITE 36 [AV :L GD 10]
> SL
> CENTRO
> FIN

```

PALOS realiza ocho llamadas al procedimiento PAL, que se encarga de dibujar un palo, así conseguimos los ocho palos del timón.

```

? PARA PALOS
> HAZ "C 0
> REPITE 8 [PAL :C HAZ "C :C + 45]
> FIN

```

PAL, como ya hemos dicho, dibuja un palo.

```

? PARA PAL :C
> PONRUMBO :C
> SL
> PONPOS [0 0]
> AV 20
> BL AV 60
> GD 90
> AV 5 GD 90
> AV 60
> BL
> AV 20
> FIN

```

Obtenemos el dibujo del timón ejecutando el procedimiento DIBUJO.

? DIBUJO

3:

En primer lugar, definimos el procedimiento A, que es el que realiza la petición de una palabra de cuatro letras:

```

? PARA A
> PONCURSOR [0 3]
> ES [INTRODUCE UNA PALABRA DE CUATRO LETRAS]
> HAZ "C1 LC
> ES :C1
> HAZ "C2 LC
> ES :C2
> HAZ "C3 LC
> ES :C3
> HAZ "C4 LC
> ES :C4
> HAZ "A ( LISTA :C1 :C2 :C3 :C4 )
> TECLA
> FIN

```

Para introducir la palabra realizaremos cuatro lecturas de carácter (una por cada le-

El efecto inverso consiste en intercambiar los colores del fondo de texto y del texto.

EXPERIENCIA Y PRACTICAS EN LOGO

tra) y las vamos asignando a diferentes variables. Una vez que tenemos las cuatro letras, transformamos en una lista las cuatro variables para poder más adelante trabajar con ella.

Al final llamamos al procedimiento TECLA, que es el que nos permite pasar a la pantalla siguiente cuando pulsamos una tecla.

El siguiente procedimiento nos presenta en la pantalla la palabra introducida:

```
? PARA B
> PONCURSOR [5 4]
> ES [LA PALABRA INTRODUCIDA
ES:]
> PONCURSOR [17 6]
> ES :A
> TECLA
> FIN
```

Una vez que nos ha mostrado en la pantalla la palabra que hemos introducido, tenemos que escribir en la parte central y en forma de cruz cada uno de los elementos que la componen. Esto lo conseguimos con el procedimiento C:

```
? PARA C
> PONCURSOR [19 9]
> ES ELEMENTO 1 :A
> PONCURSOR [21 11]
> ES ELEMENTO 2 :A
> PONCURSOR [19 13]
> ES ELEMENTO 3 :A
> PONCURSOR [17 11]
> ES ELEMENTO 4: A
> ESPERA 15
> FIN
```

Para que la palabra vaya rotando, lo que hacemos es transformar la palabra que tenemos en otra, en la cual el último elemento desaparece de su posición y pasa a ser el primero.

```
? PARA D
> HAZ "A MU PP (ULTIMO :A) :A
> FIN
```

Por ejemplo, si la palabra introducida es LOGO, la lista la primera vez tendrá sus elementos de la siguiente forma:

LOGO

La primera vez que se ejecute el pro-

cedimiento D, sus elementos aparecerán en la siguiente posición:

OLOG

Una vez que hemos cambiado de posición los elementos de la lista, lo único que tenemos que hacer es volver a escribir la lista mediante el procedimiento B.

El procedimiento que escribe el mensaje PULSA UNA TECLA y que realiza una pausa hasta que una tecla que se haya pulsado es:

```
? PARA TECLA
> PONCURSOR [12 22]
> ES [PULSA UNA TECLA]
> HAZ "T LC
> BP
> BT
> FIN
```

Por último, definimos el procedimiento que inicializa el proceso y engloba a todos los restantes:

```
? PARA TODO
> BP
> OT
> PT
> BT
> A
> B
> REPITE 100 [C D]
> FIN
```

Para ejecutar el proceso introducimos:

```
? TODO
```

4:

Para realizar el dibujo definimos los siguientes procedimientos:

```
? PARA CABEZA
> PONCURSOR [15 5]
> ES "ZZZZZZZZZ
> HAZ "C 15
> REPITE 2 [HAZ "F 6 REPITE 8 [PON-
CURSOR LISTA :C :F ES "Z HAZ "F :F
+ 1] HAZ "C 23]
> PONCURSOR [15 14]
> ES "ZZZZZZZZZ
> FIN
```

La pantalla del SPECTRUM dispone de un borde al cual se le puede cambiar el color.

? PARA OJOS
 > PONCURSOR [17 8]
 > ES ">
 > PONCURSOR [21 8]
 > ES "<
 > FIN

? PARA NARIZ
 > PONCURSOR [19 10]
 > ES "Y
 > FIN

? PARA BOCA
 > PONCURSOR [19 12]
 > ES "="
 > FIN

? PARA OREJAS
 > PONCURSOR [14 9]
 > ES "S
 > PONCURSOR [24 9]
 > ES "S
 > FIN

Ahora los englobamos todos en CARA:

? PARA CARA
 > BP
 > BT
 > OT
 > CABEZA
 > OJOS
 > NARIZ
 > BOCA
 > OREJAS
 > FIN

Para verlo bastará con introducir:

? CARA

5:

— ? PARA A1
 > HAZ "A PONPROP "A1 "SALUDOS
 "HOLA
 > ES :A
 > FIN

INCORRECTO

Si ejecutamos este procedimiento, el LOGO devolverá el mensaje de error:

PONPROP NO HA DEVUELTO NADA A
 HAZ

No podemos guardar en una variable una asignación de propiedad.

— ? PARA A2
 > PONPROP "A2 "SALUDOS "ADIOS
 > ES PROP "A2 "SALUDOS
 > FIN

INCORRECTO

El nombre especificado en la primitiva PROP es una variable e la que no hemos asignado ningún valor, sería correcto si en vez de indicar el nombre A2 como variable lo hiciésemos como nombre de procedimiento.

? ES PROP "A2 "SALUDOS

— ? PARA A3
 > PONPROP "A3 "SALUDOS "ADIOS
 > BOPROP "A3 "SALUDOS
 > FIN

CORRECTO

Primero asociamos al procedimiento A3 la propiedad SALUDOS con el valor ADIOS, y luego la borramos.

— ? PARA EFECTOS
 > PONBRILLO 1
 > FLASH
 > INVERSO
 > ESC "EFECTOS
 > FIN

CORRECTO

La palabra EFECTOS aparecerá en la pantalla con brillo, el flash y en inverso.

— ? FLASH 1
 ? PONCB 2
 ? PONCF 3

INCORRECTO

La primera orden que se introduce (FLASH 1) no puede llevar parámetro.

— ? NORMAL

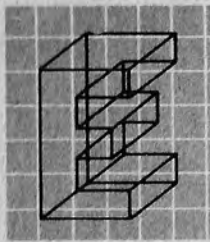
CORRECTO

La orden NORMAL se puede introducir sin que esté activado el brillo o el inverso. En este caso su acción será nula.

Puedes conseguir bonitos efectos con las posibilidades que nos ofrece el SPECTRUM.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

La alta resolución en el MSX



El MSX es uno de los ordenadores con mayor capacidad gráfica del mercado. En él podemos disponer de dos tipos de pantalla, distintos ambos, con la posibilidad de utilizar los 126 colores de que disponemos. Estas son:

SCREEN 2. Los gráficos se realizan

con unidades de un punto. Disponemos de 256×192 puntos en la pantalla.

SCREEN3. Los gráficos se realizan con unidades de 4×4 puntos. Disponemos de 256×192 puntos en la pantalla, pero como están agrupados de cuatro en cuatro, la resolución se nos reduce a 64×48 puntos. El primer programa que vamos a ver nos dibuja una serie de líneas. Con este programa pretendemos mostrar la diferencia entre ambos tipos de pantalla.

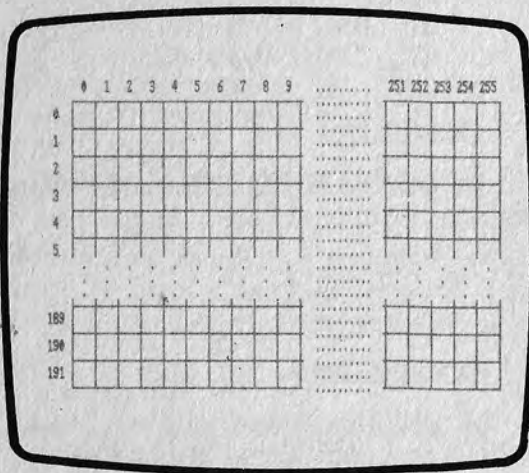


Fig. 1. Resolución en SCREEN 2.

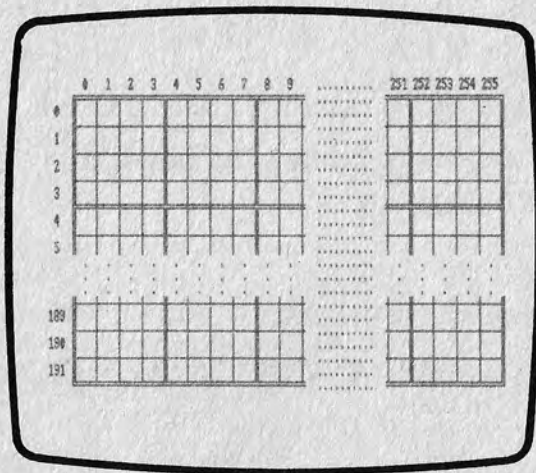
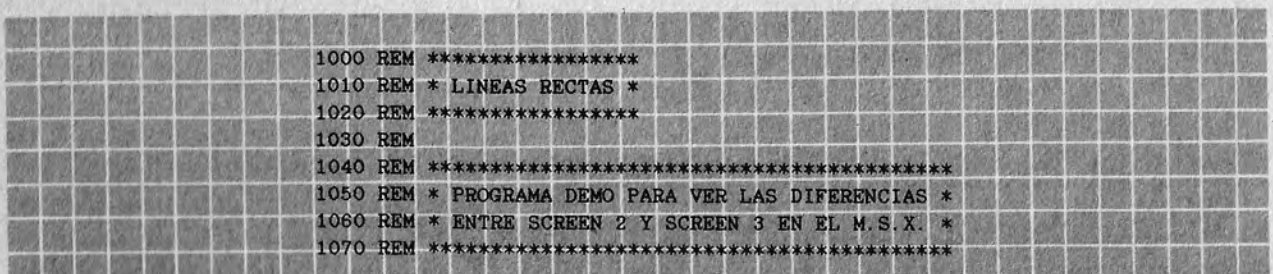


Fig. 2. Resolución en SCREEN 3. Obsérvese que los puntos físicos están agrupados en una matriz de 4×4 puntos lógicos



```

1080 REM
1090 REM *** FUNCIONAMIENTO EN SCREEN 2 ***
1100 REM
1110 SCREEN 2
1120 CLS
1130 OPEN "GRP:" FOR OUTPUT AS #1
1140 PSET (0,0)
1150 PRINT #1,"SCREEN 2"
1160 LET CO=2
1170 FOR I=10 TO 190 STEP 10
1180     LINE (0,I)-(255,I),CO
1190     LET CO=CO+1
1200     IF CO=16 THEN LET CO=2
1210 NEXT I
1220 CLOSE #1
1230 FOR I=1 TO 2000
1240 NEXT I
1250 REM
1260 REM *** FUNCIONAMIENTO EN SCREEN 3 ***
1270 REM
1280 SCREEN 3
1290 CLS
1300 OPEN "GRP:" FOR OUTPUT AS #1
1310 PSET (0,0)
1320 PRINT #1,"SCREEN 3"
1330 LET CO=2
1340 FOR I=40 TO 190 STEP 10
1350     LINE(0,I)-(255,I),CO
1360     LET CO=CO+1
1370     IF CO=16 THEN LET CO=2
1380 NEXT I
1390 CLOSE #1
1400 FOR I=1 TO 2000
1410 NEXT I
1420 REM
1430 REM *** VUELTA A EMPEZAR ***
1440 REM
1450 RUN

```

La figura 1 nos muestra cómo se vería la ejecución de este programa en la pantalla de nuestro ordenador.

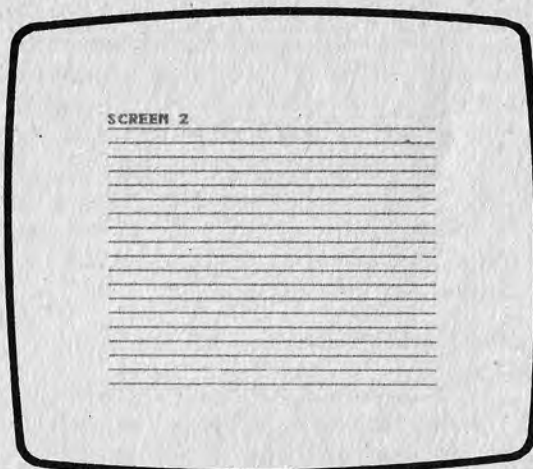


Fig. 3. Ejecución del programa 1.

Como habrá visto el lector, en el programa aparecen algunas instrucciones que no hemos estudiado. Estas serán las que veremos en este tomo, pero antes de hacerlo hay que decir unas cuantas cosas más sobre la alta resolución.

Por lo general, a la hora de dibujar nuestros propios gráficos en el MSX utilizaremos el modo 2 de pantalla, ya que el 3 tiene muy poca resolución para poder conseguir resultados aceptables. Por ello, hay que decir, antes de comenzar, que en este ordenador cada punto no puede ser de un color, sino que los colores se agrupan en grupos de cuatro puntos.

Esto quiere significar lo siguiente: si dibujamos un punto de color rojo en la posición 0,0 de la pantalla y otro de color verde en la posición 1,0, entonces el color que estaba rojo se volverá verde. Si ahora ponemos de color

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

azul el punto 2,0, entonces los dos puntos anteriores se pondrán azules y si, por fin, ponemos el punto 3,0 de color blanco, los cuatro estarán blancos. Sin embargo, si ahora ponemos el color 4,0 de color violeta, los otros cuatro puntos continuarán de color blanco. Esto significa que cada punto no puede ser de un color cualquiera, sino que tendrá el mismo color que los otros tres puntos del grupo de cuatro sobre el que se encuentra.

Esta regla sólo sirve horizontalmente, pues si ponemos el punto 0,0 de color azul y el punto 0,1 de color verde, el 0,0 no cambiará a verde, sino que continuará de color azul. Por ello, los puntos horizontales se encuentran en grupos de cuatro y los verticales en solitario.

Los comandos de alta resolución en el MSX

Las instrucciones PSET y PRESET

Con esta instrucción podremos iluminar puntos en la pantalla con sólo darle las coordenadas del punto a iluminar. Su sintaxis es la siguiente:

PSET (X,Y),C

donde:

X = Coordenada en X en la pantalla del punto a iluminar.

Y = Coordenada en Y en la pantalla del punto a iluminar.

C = Color de dicho punto. Se admite un número entre 0 y 15. Cualquier otro nos dará error.

Con el programa 2 podemos ver cómo funciona esta instrucción.

```

1000 REM *****
1010 REM * NOCHE ESTRELLADA *
1020 REM *****
1030 REM
1040 REM *****
1050 REM * PROGRAMA DEMOSTRACION PARA VER EL *
1060 REM * USO DEL PSET Y PRESET EN EL MSX. *
1070 REM *****
1080 REM
1090 SCREEN 2
1100 CLS
1110 OPTION BASE 1
1120 DIM XX(500)
1130 DIM YY(500)
1140 REM
1150 REM *** DIBUJO DEL CIELO ESTRELLADO ***

```

```

1180 REM
1170 OPEN "GRP:" FOR OUTPUT AS #1
1180 PRESET(0,0)
1190 PRINT #1, "DIBUJANDO EL CIELO";
1200 LET A=RND(-TIME)
1210 FOR I=1 TO 500
1220 LET X=INT(RND(1)*256)
1230 LET Y=INT(RND(1)*192)
1240 LET C=INT(RND(1)*16+1)
1250 IF C<2 THEN GOTO 1240
1260 PSET(X,Y),C
1270 LET XX(I)=X
1280 LET YY(I)=Y
1290 NEXT I
1300 REM
1310 REM *** BUCLE DE RETARDO ***
1320 REM
1330 FOR I=1 TO 1500
1340 NEXT I
1350 REM
1360 REM *** BORRADO DEL CIELO ESTRELLADO ***
1370 REM
1380 BEEP:BEEP
1390 PRESET(0,8)
1400 PRINT#1, "BORRANDO EL CIELO ";
1410 FOR I=1 TO 500
1420 LET X=XX(I)
1430 LET Y=YY(I)
1440 PRESET (X,Y)
1450 NEXT I
1460 GOTO 1460

```

Tenemos otra instrucción que nos permite borrar un punto de la pantalla y ponerlo del mismo color que el del papel. Esta es la instrucción PRESET, cuya sintaxis es la siguiente:

PRESET (X,Y)

donde X e Y son las coordenadas del punto en la pantalla.

Si el lector estudia con atención el programa 2, podrá ver que todos los puntos que pusimos con la instrucción PSET más tarde los borramos con PRESET, independientemente del color de cada punto.

Hay que pensar que si tuviésemos que dibujar líneas rectas y curvas con PSET, cualquier dibujo que intentásemos hacer tardaría mucho tiempo en realizarse. Por ello, tenemos otra serie de instrucciones para conseguir pintar círculos, arcos y rectas.

La instrucción LINE

Con esta instrucción podemos dibujar líneas rectas entre dos puntos cualesquiera de la pantalla. La sintaxis es la siguiente:

LINE (X1,Y1)-(X2,Y2),C

donde:

X1 e Y1 son las coordenadas del punto inicial.

X2 e Y2 son las coordenadas del punto final.

C es el color que queremos que tenga la recta.

También tenemos otra forma de escribir esta instrucción:

LINE -(X2,Y2),C

Esta variante sirve para dibujar una línea recta desde el último punto o línea trazada, no importa cuál sea, hasta el punto que tiene como coordenadas a X2 e Y2. En el programa 1 vimos cómo utilizar esta instrucción. Se recomienda al lector que intente dibujar algunas líneas y algunas formas geométricas para entender mejor su manejo.

Pero esta instrucción no sólo nos puede servir para dibujar rectas, sino que con ella

podemos dibujar cuadrados y rectángulos. Podemos dibujarlos sólo con su contorno o rellenos de color. Para hacerlo sólo tenemos que añadir una coma seguida de una B (BOX) si queremos ver los límites del rectángulo, o de BF (BOX FILLED) si lo queremos ver relleno de color.

LINE (X1,Y1)-(X2,Y2),C,B
LINE (X1,Y1)-(X2,Y2),C,BF

En este caso especial las coordenadas que están entre paréntesis tienen una función distinta a la anterior. Ahora las coordenadas X1 e Y1 nos indican el punto en el cual estará la esquina superior izquierda del rectángulo, y las coordenadas X2, Y2 el punto que servirá como vértice para la esquina inferior derecha del rectángulo.

El programa que vamos a ver a continuación nos muestra las diferencias entre un rectángulo del cual sólo vemos sus contornos y de otro relleno de color. También nos permite ver cómo funciona la instrucción LINE.

```
1000 REM *****
1010 REM * NEW YORK *
1020 REM *****
1030 REM
1040 REM *****
1050 REM * PROGRAMA DEMOSTRACION PARA VER EL USO DEL *
1060 REM * COMANDO LINE EN EL M.S.X. *
1070 REM *****
1080 REM
1090 REM *** INICIALIZACION Y DIBUJO DE LA PANTALLA ***
1100 REM
1110 SCREEN 2
1120 CLS
1130 OPEN "GRP:" FOR OUTPUT AS #1
1140 PRESET (0,170)
1150 PRINT#1,"VISTA AEREA DE NEW YORK";
1160 LINE (0,0)-(255,120),15,B
1170 LINE (2,2)-(253,118),15,B
1180 LET XMIN=15
1190 LET YMIN=15
1200 LET XMAX=235
1210 REM
1220 REM *** RECOGIDA DE OPCION ***
1230 REM
1240 LET A=RND(-TIME)
1250 PRESET (0,152)
1260 PRINT#1,"COMO QUIERES LOS EDIFICIOS?"
1270 PRINT#1,"(L)LENOS O (V)ACIOS)"
1280 LET A$=INKEY$
1290 IF A$="" THEN GOTO 1280
1300 IF A$="L" OR A$="l" THEN LET SW=1:GOTO 1340
1310 IF A$="V" OR A$="v" THEN LET SW=0:GOTO 1340
1320 GOTO 1280
1330 PRESET (0,175)
```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

1340 PRINT#1, "PULSA 'P' PARA PARAR"
1350 REM
1360 REM *** DIBUJO DE NEW YORK ***
1370 REM
1380 LET X1=INT(RND(1)*XMAX+XMIN)
1390 LET Y1=INT(RND(1)*100+YMIN)
1400 LET X2=INT(RND(1)*XMAX+XMIN)
1410 LET Y2=118
1420 LET C=INT(RND(1)*14+3)
1430 IF ABS(X2-X1)>80 THEN GOTO 1380
1440 IF SW=0 THEN LINE(X1,Y1)-(X2,Y2),C,B
1450 IF SW=1 THEN LINE(X1,Y1)-(X2,Y2),C,BF
1460 REM
1470 REM *** LECTURA DEL TECLADO Y RETARDO ***
1480 REM
1490 FOR I=1 TO 100
1500 IF INKEY$="P" THEN GOTO 1530
1510 NEXT I
1520 GOTO 1380
1530 CLS
1540 CLOSE #1
1550 END

```

La instrucción CIRCLE

Aunque esta instrucción nos recuerde al círculo, sirve para más cosas que para dibujar círculos. También nos permitirá dibujar arcos de circunferencia y elipses con sólo darle una serie de parámetros. Su sintaxis general es:

CIRCLE (X,Y),R,C,A1,A2,S

donde:

X = Coordenada en X del centro del círculo.

Y = Coordenada en Y del centro del círculo.

C = Color del círculo.

R = Radio (en pixels) del círculo.

A1 = Angulo del que partimos para dibujar.

A2 = Angulo al que llegamos dibujando.

S = Relación entre el radio horizontal y el vertical.

Los cuatro primeros parámetros están claros. Sólo tenemos que explicar un poco más los tres restantes.

Si incluimos los parámetros A1 y A2 en la instrucción tendremos la posibilidad de dibujar arcos de circunferencia. Estos dos valores le dicen al ordenador cuál es el ángulo desde el que comenzará a dibujar y cuál el ángulo donde debe terminar. Dichos ángulos han de estar medidos en radianes.

El parámetro S le dice al ordenador la relación existente entre el radio vertical y el horizontal. En un círculo dicha relación es igual a uno, pero si hacemos que sea igual a 0,5, entonces el radio vertical será el doble que el horizontal y, por tanto, en la pantalla nos aparecerá una elipse.

El programa que vamos a ver a continuación nos permite ver cómo se dibujan una serie de círculos (o elipses, según elija el lector) por la pantalla.

```

1000 REM *****
1010 REM * ESPIRALES *
1020 REM *****
1030 REM
1040 REM *****
1050 REM * PROGRAMA DE DEMOSTRACION PARA VER EL USO *
1060 REM * DEL COMANDO CIRCLE EN EL M.S.X. *
1070 REM *****

```

```

1080 REM
1090 REM *** INICIALIZACION ***
1100 REM
1110 LET XCEN=160
1120 LET YCEN=100
1130 SCREEN 0
1140 CLS
1150 KEY OFF
1160 REM
1170 REM *** ENTRADA DE OPCION ***
1180 REM
1190 LOCATE 1,10
1200 PRINT "PULSA UN NUMERO ENTRE 3 Y 9!"
1210 LET A$=INKEY$
1220 IF A$="" THEN GOTO 1210
1230 IF A$<"3" OR A$>"9" THEN GOTO 1210
1240 LET N=VAL(A$)
1250 LOCATE 1,13
1260 PRINT "QUE QUIERES?"
1270 PRINT
1280 PRINT "1 - CIRCULOS (S=1.0)"
1290 PRINT "2 - ELIPSES (S=0.5)"
1300 PRINT "3 - ELIPSES (S=2.0)"
1310 LET A$=INKEY$
1320 IF A$="" THEN GOTO 1310
1330 IF A$<"1" OR A$>"3" THEN GOTO 1310
1340 IF A$="1" THEN LET S=1
1350 IF A$="2" THEN LET S=.5
1360 IF A$="3" THEN LET S=2
1370 REM
1380 REM *** PROGRAMA PRINCIPAL ***
1390 REM
1400 SCREEN 2
1410 CLS
1420 FOR I=0 TO -99 STEP -.2
1430 LET X=XCEN+I*N*COS(I)
1440 LET Y=YCEN+I*N*SIN(I)
1450 CIRCLE(X,Y),10,,,S
1460 NEXT I
1470 FOR I=-1 TO 4000
1480 NEXT I
1490 RUN

```

Referente al uso de los argumentos A1 y A2, en la sentencia CIRCLE no es arbitrario su orden, ya que si, por ejemplo, decimos:

CIRCLE (100,100),30,15,0,3.14

el resultado será muy distinto del que obtendríamos si ponemos:

CIRCLE (100,100),30,15,3.14,0

Para probar esto, aparece a continuación la figura 4:

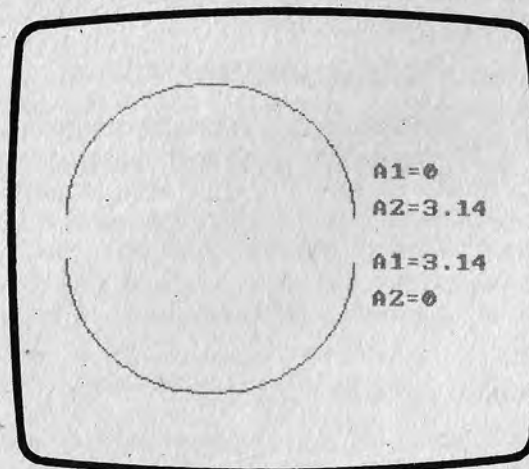


Fig. 4. Diferencia del dibujo según el orden de colocación de los ángulos.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

La instrucción PAINT

Esta potente instrucción nos va a permitir colorear zonas cerradas de una forma rápida y perfecta. Lo único que es realmente necesario es que la superficie que queremos rellenar sea totalmente cerrada. Con esta instrucción podemos rellenar círculos, cuadrados, triángulos, cualquier tipo de polígono e incluso figuras irregulares, por complicadas que sean. Su sintaxis es la siguiente:

PAINT (X,Y),CR,CL

donde:

X e Y son las coordenadas desde donde queremos que parta el relleno. En la práctica dicha posición no es importante. Lo que sí es realmente necesario es que se encuentre dentro de la figura a rellenar.

CR es el color con que queremos rellenar la figura.

CL es el color del borde o límite. Es el color que tienen las líneas que delimitan las figuras.

Como este comando es muy sencillo de entender, no necesita la realización de un programa para ver su funcionamiento. Aun así, si cambiamos una línea del programa 3, cuando el programa nos dibuje rectángulos rellenos de color, no lo hará por el parámetro BF, sino por una instrucción PAINT. Esta línea es:

```
1450 IF SW=1 THEN LINE(X1,Y1)-(X2,Y2),C,B:
      PAINT(X1+1,Y1+1),C,C
```

La instrucción POINT

Esta es una de las instrucciones más útiles y que más utilizaremos en el caso de que queramos realizar movimientos en alta resolución. Con ella podremos saber de qué color es un determinado punto de la pantalla. Su sintaxis no puede ser más sencilla:

LET var=POINT(X,Y)

El resultado es siempre un número entre 0 y 15, por lo que podemos hacer con dicho número lo que hacemos con cualquier otro. El número que nos devuelve es el número que corresponde con el color de dicho punto en la pantalla. No hay que decir que X e Y son las coordenadas de dicho punto en la pantalla.

Especificaciones de STEP. Usos

Hasta ahora las coordenadas que le dábamos a todas las instrucciones que hemos vis-

to eran coordenadas absolutas de la pantalla. Esto significa que si estábamos en la posición 100, 100 y queríamos movernos diez posiciones a la derecha y diez posiciones hacia abajo, teníamos que decirle a la instrucción LINE que se moviese hacia la 110, 110. Pero a veces no sabemos la dirección absoluta en la pantalla, a la cual nos queremos mover, pero sí sabemos la dirección relativa. Para solventar este problema podemos utilizar la orden STEP dentro de LINE, para decir que vamos a hablar de coordenadas relativas al punto en el que nos encontramos.

Este subcomando puede utilizarse en los comandos CIRCLE, LINE, PAINT, PSET, PRESET e, incluso, en PUT SPRITE. Por ejemplo, si nos encontramos en la posición 100, 100 y queremos dibujar un círculo de radio 14.2 con un incremento en X y en Y de 10 puntos, sólo tenemos que escribir:

CIRCLE STEP (10,10),14.2

Según esto, se puede ver que el subcomando STEP hay que ponerlo tras la instrucción a la que queremos darle coordenadas relativas. Otro ejemplo lo podemos ver con la instrucción PAINT:

PAINT SEPT (20,20),12,14

La macroinstrucción DRAW en el MSX y en el IBM

Los ordenadores MSX e IBM disponen de una macroinstrucción de uso múltiple que nos permite realizar las mismas funciones que hemos visto, a excepción de CIRCLE, a una velocidad mayor y ahorrando mucha memoria. La sintaxis de esta instrucción es la siguiente:

DRAW submandato

El submandato puede ser una cadena de caracteres especiales que veremos a continuación o una variable alfanumérica. Estos caracteres especiales se componen de letras y números que, dependiendo de cuáles sean, le dicen al ordenador qué función han de realizar. Estas letras son:

Mx,y - (MOVEx,y) Dibuja una línea desde la última posición en que se dibujó hasta la posición x,y de la pantalla en coordenadas absolutas. Si ponemos una B mayúscula antes de la M, entonces el cursor gráfico se moverá a la posición x,y sin dibujar. Por ejemplo, la siguiente línea nos dibuja una recta desde la posición 100, 100 a la 123, 150.

DRAW "BM100,100 M123,130"

Una vez que se ha dibujado la línea, el cursor gráfico queda en la posición 123,130, por lo que si queremos dibujar otra línea desde esta posición a la 40, 100, tendremos que escribir:

`DRAW "BM100,100 M123,130 M40,100"`

Pero hay veces en que necesitamos que el cursor gráfico tras dibujar una línea, se vuelva a colocar en su posición antigua. Por ejemplo, si queremos dibujar una línea desde 100, 100 a 120, 120, y otra línea desde 100, 100 a 30, 10. Para ello podemos poner una N antes de la M. Con ello, tras dibujar la línea que se le indica, el cursor volverá a la posición desde la que partió. Estas dos líneas se escribirían como:

`DRAW "BM100,100 NM120,120 NM30,10"`

Con todo lo dicho, cada vez que dibujamos una línea con el subcomando M, le estamos dando las coordenadas absolutas de la pantalla. Al igual que ocurre con la instrucción LINE, hay veces que no sabemos la coordenada absoluta, pero sí sabemos cuánto nos queremos mover desde la posición en la que nos encontramos. Para ello podemos mover, antes de la coordenada a la que nos queremos mover, un signo más (+) o menos (-) para indicar que el movimiento ha de ser relativo y en qué dirección. Por ejemplo, si estamos en una posición no conocida y queremos movernos ocho puntos a la izquierda y 12 hacia abajo, sólo tendremos que poner:

`DRAW "M-8,+12"`

Un (UP n). Nos dibuja una línea hacia arriba de **n** partes o puntos.

Dn (DOWN n). Nos dibuja una línea hacia abajo de **n** partes o puntos.

Rn (RIGHT n). Nos dibuja una línea hacia la derecha de **n** partes o puntos.

Ln (LEFT n). Nos dibuja una línea hacia la izquierda de **n** parte o puntos.

Estas cuatro instrucciones son más rápidas que si dibujamos con M y nos permiten crear cuadrados, rectángulos y, en general, cualquier figura cuyos ángulos sean todos rectos. Por ejemplo, para dibujar un cuadrado partiendo de la posición 100, 100, podríamos escribir:

`DRAW "BM100,100 U20 R20 D20 L20"`

En. Nos dibuja una línea de **n** partes o puntos con un ángulo de 45 grados con respecto a la parte positiva del eje de X.

Fn. Nos dibuja una línea de **n** pasos o puntos con un ángulo de 135 grados con respecto a la parte positiva del eje de X.

Gn. Nos dibuja una línea de **n** pasos o puntos con un ángulo de 225 grados con respecto a la parte positiva del eje de X.

Hn. Nos dibuja una línea de **n** pasos o puntos con un ángulo de 315 grados con respecto a la parte positiva del eje de X.

Para entender mejor estas ocho direcciones en las que podemos dibujar con facilidad, vemos la figura 5.

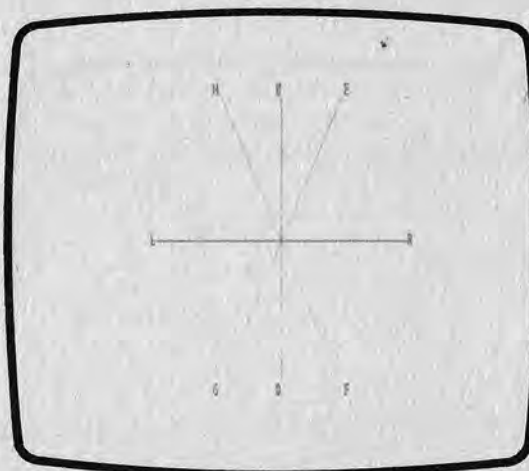


Fig. 5. Distintas direcciones que se podrán usar con la instrucción DRAW.

En estas ocho nuevas instrucciones también podemos colocar antes de la letra que la representa una B, para movernos sin dibujar, o una N para dibujar volviendo al punto de partida.

Cn (COLOR n). Sirve para cambiar el color del lápiz con el que nos imaginamos que estamos dibujando.

Sn (SCALE N). Nos sirve para variar la escala con la que dibujamos. **n** será un número entre 1 y 255. Por defecto es 4. Cada valor representa 1/4 de pixel. Por ello, si decimos que:

`DRAW "BM100,100 S4 U10"`

el dibujo resultante será una línea que parte del pixel 100,100 y que sube hasta el pixel 100,90. Se ha realizado una línea hacia arriba de 10 pixels. Sin embargo, si ponemos:

`DRAW "BM100,100 S2 U10"`

como hemos variado la escala a la mitad, la lí-

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

nea dibujada será sólo de 5 pixels, y si ponemos:

```
DRAW "BM100,100 S8 U10"
```

la línea será de 20 pixels.

An (ANGULO n). Nos permite hacer girar el sistema de coordenadas de referencia. Con este subcomando podemos dibujar objetos boca arriba, boca abajo, mirando a la derecha o a la izquierda sin cambiar la instrucción. **n** sólo puede tener valores comprendidos entre 0 y 3 y el resultado del giro del eje de coordenadas para cada uno de estos cuatro valores puede verse en la figura 6.

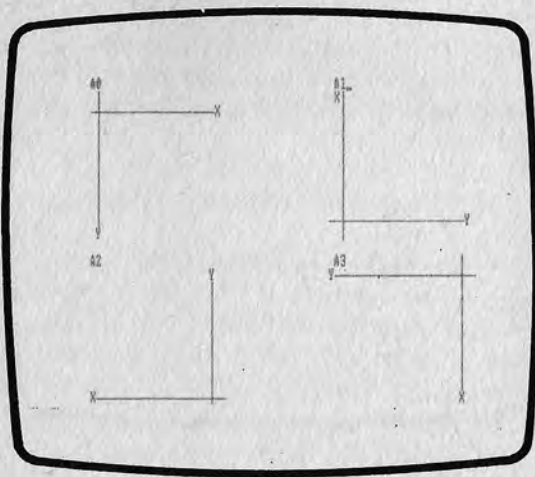


Fig. 6. Distintos valores de A y su efecto.

Dijimos que el submandato podía ser una cadena de caracteres o una variable alfanumérica. Esto es así porque el ordenador nos acepta igualmente:

```
DRAW "BM100,100 U10"
```

que:

```
DRAW A$
```

si A\$="BM100,100 U10"

Xvaralf. Nos permite incluir dentro de una cadena otra subcadena que realiza una cierta función. Sirve para incluir dentro de las comillas los comandos de otra variable alfanumérica para ahorrar memoria. Un ejemplo de utilización sería:

```
DRAW "BM100,100 XA$; BM110,110 XA$"
```

donde A\$="U20 R20 D20 L20"

Este subcomando nos sirve para repetir acciones varias veces sin tener que teclearlas cada vez que las necesitamos. Podemos pensar que este comando permitiese el uso de subrutinas dentro del comando DRAW. Es muy importante no olvidar poner el punto y coma (;) tras el signo dólar (\$) de la variable alfanumérica.

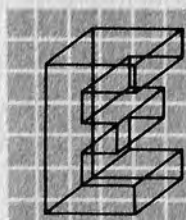
Por otro lado, hay veces en que necesitamos utilizar variables numéricas para hacer un dibujo. Si quisiésemos llenar la pantalla de cuadrados sería demencial intentar escribir veinte o treinta líneas para dibujar cada uno de los cuadrados. Como hay veces que el argumento numérico que queremos darle a instrucciones como S,C,A,D,U,L,R o M queremos que varíe, para que el dibujo pueda variar, podemos incluir una variable numérica tras estas instrucciones. Con ello, la acción a desarrollar cogerá el valor de dicha variable como argumento a su función. La sintaxis es la siguiente:

```
DRAW "(COMANDO A UTILIZAR)=var; ..."
```

como, por ejemplo:

```
LET SS=8:DRAW "S=SS; BM100,100 U20"
```

TRUCOS Y RUTINAS BASICAS



N tomos anteriores vimos un programa que nos permitía realizar SCROLL en las cuatro direcciones de la pantalla con los atributos (colores) de ésta. A continuación vamos a ver un programa que nos permitirá realizar dicho

SCROLL, pero con los caracteres de la pantalla.

Esta rutina, para SPECTRUM, conjugada con la que ya hemos visto, nos permitirá hacer que nuestros juegos sean mucho más vivos y rápidos.

```
1000 REM *****
1005 REM *
1010 REM * SCROLL DE CARACTERES *
1015 REM *
1020 REM *****
1030 REM
1040 REM *****
1045 REM *
1050 REM * PROGRAMA DEMOSTRACION *
1055 REM *
1060 REM *****
1070 REM
1090 REM *****
1100 REM *
1110 REM * (c) Ediciones Siglo Cultural *
1120 REM *
1130 REM * (c) 1987
1140 REM *
1150 REM *****
1160 REM
1170 REM *****
1180 REM * INICIALIZACION *
1190 REM *****
1200 REM
1210 PAPER 0
1220 INK 7
1230 BORDER 0
1240 CLS
1250 CLEAR 55499
1251 PRINT AT 10,8: INK 4: FLASH 1: "CARGANDO DATAS"
1252 PRINT AT 12,6: INK 5: FLASH 1: INVERSE 1: "ESPERE UN MOMENTO."
1255 REM
1260 REM *****
1270 REM * LECTURA DE LAS LINEAS DATA *
1280 REM *****
1290 REM
```

TRUCOS Y RUTINAS BASICAS

```
1300 LET TOT=0
1310 LET LIN=9040
1320 FOR I=55500 TO 55889 STEP 15
1330   LET CHEK=0
1340   LET LIN=LIN+10
1350   FOR J=I TO I+14
1360     READ A
1370     LET CHEK=CHEK+A
1380     POKE J,A
1390   NEXT J
1400   READ TOT
1410   IF TOT<>CHEK THEN GO TO 5000
1420 NEXT I
2000 REM
2010 REM *****
2020 REM *
2030 REM * DEMOSTRACION *
2040 REM *
2050 REM *****
2060 REM
2070 REM *****
2080 REM * SCROLL A LA IZQUIERDA *
2090 REM *****
2100 REM
2110 CLS
2120 POKE 23300.1 : REM SCROLL A LA IZQUIERDA
2130 POKE 23301.1 : REM LA PARTE SUPERIOR (1)
2140 REM
2145 PRINT AT 0.0:
2150 GO SUB 8000
2160 FOR I=0 TO 31
2170   RANDOMIZE USR 55500
2180   PAUSE 4
2190 NEXT I
2200 REM
2210 POKE 23301.2 : REM LA PARTE CENTRAL (2)
2220 REM
2225 PRINT AT 0.0:
2230 GO SUB 6000
2240 FOR I=0 TO 31
2250   RANDOMIZE USR 55500
2260   PAUSE 4
2270 NEXT I
2280 REM
2290 POKE 23301.3 : REM LA PARTE INFERIOR (3)
2300 REM
2310 PRINT AT 0.0:
2315 GO SUB 8000
2320 FOR I=0 TO 31
2330   RANDOMIZE USR 55500
2340   PAUSE 4
2350 NEXT I
2360 REM
2370 POKE 23301.4 : REM LAS DOS PRIMERAS PARTES (1 Y 2)
2380 REM
2390 PRINT AT 0.0:
2400 GO SUB 6000
2410 FOR I=0 TO 31
2420   RANDOMIZE USR 55500
2430   PAUSE 4
2440 NEXT I
2450 REM
2460 POKE 23301.5 : REM LAS DOS ULTIMAS PARTES (2 Y 3)
2470 REM
2480 PRINT AT 0.0:
2490 GO SUB 8000
2500 FOR I=0 TO 31
```

```

2510    RANDOMIZE USR 55500
2520    PAUSE 4
2530 NEXT I
2540 REM
2550 POKE 23301.6 : REM TODA LA PANTALLA (1,2 Y 3)
2560 REM
2570 PRINT AT 0,0;
2580 GO SUB 6000
2590 FOR I=0 TO 31
2600    RANDOMIZE USR 55500
2610    PAUSE 4
2620 NEXT I
2630 REM
2640 REM *****
2650 REM * SCROLL A LA DERECHA *
2660 REM *****
2670 REM
2680 POKE 23300.2 : REM SCROLL A LA DERECHA
2690 POKE 23301.1 : REM LA PARTE SUPERIOR (1)
2700 REM
2710 PRINT AT 0,0;
2720 GO SUB 6000
2730 FOR I=0 TO 31
2740    RANDOMIZE USR 55500
2750    PAUSE 4
2760 NEXT I
2770 REM
2780 POKE 23301.2 : REM LA PARTE CENTRAL (2)
2790 REM
2800 PRINT AT 0,0;
2810 GO SUB 6000
2820 FOR I=0 TO 31
2830    RANDOMIZE USR 55500
2840    PAUSE 4
2850 NEXT I
2860 REM
2870 POKE 23301.3 : REM LA PARTE INFERIOR (3)
2880 REM
2890 PRINT AT 0,0;
2900 GO SUB 6000
2910 FOR I=0 TO 31
2920    RANDOMIZE USR 55500
2930    PAUSE 4
2940 NEXT I
2950 REM
2960 POKE 23301.4 : REM LAS DOS PRIMERAS PARTES (1 Y 2)
2970 REM
2980 PRINT AT 0,0;
2990 GO SUB 6000
3000 FOR I=0 TO 31
3010    RANDOMIZE USR 55500
3020    PAUSE 4
3030 NEXT I
3040 REM
3050 POKE 23301.5 : REM LAS DOS ULTIMAS PARTES (2 Y 3)
3060 REM
3070 PRINT AT 0,0;
3080 GO SUB 6000
3090 FOR I=0 TO 31
3100    RANDOMIZE USR 55500
3110    PAUSE 4
3120 NEXT I
3130 REM
3140 POKE 23301.6 : REM TODA LA PANTALLA (1,2 Y 3)
3150 REM
3160 PRINT AT 0,0;
3170 GO SUB 6000
3180 FOR I=0 TO 31

```

TRUCOS Y RUTINAS BASICAS

```

3190 RANDOMIZE USR 55500
3200 PAUSE 4
3210 NEXT I
3220 REM
3230 REM *****
3240 REM * SCROLL HACIA ARRIBA *
3250 REM *****
3260 REM
3270 POKE 23300.3 : REM SCROLL HACIA ARRIBA
3280 POKE 23301.1 : REM LA PARTE SUPERIOR (1)
3290 REM
3300 PRINT AT 0.0:
3310 GO SUB 6000
3320 FOR I=0 TO 7
3330 RANDOMIZE USR 55500
3340 PAUSE 4
3350 NEXT I
3360 REM
3370 POKE 23301.2 : REM LA PARTE DEL CENTRO (2)
3380 REM
3390 PRINT AT 0.0:
3400 GO SUB 6000
3410 FOR I=0 TO 7
3420 RANDOMIZE USR 55500
3430 PAUSE 4
3440 NEXT I
3450 REM
3460 POKE 23301.3 : REM LA PARTE INFERIOR (3)
3470 REM
3480 PRINT AT 0.0:
3490 GO SUB 6000
3500 FOR I=0 TO 7
3510 RANDOMIZE USR 55500
3520 PAUSE 4
3530 NEXT I
3540 REM
3550 POKE 23301.4 : REM LAS DOS PRIMERAS PARTES (1 Y 2)
3560 REM
3570 PRINT AT 0.0:
3580 GO SUB 6000
3590 FOR I=0 TO 15
3600 RANDOMIZE USR 55500
3610 PAUSE 4
3620 NEXT I
3630 REM
3640 POKE 23301.5 : REM LAS DOS PARTES INFERIORES (2 Y 3)
3650 REM
3660 PRINT AT 0.0:
3670 GO SUB 6000
3680 FOR I=0 TO 15
3690 RANDOMIZE USR 55500
3700 PAUSE 4
3710 NEXT I
3720 REM
3730 POKE 23301.6 : REM TODA LA PANTALLA (1,2 Y 3)
3740 REM
3750 PRINT AT 0.0:
3760 GO SUB 6000
3770 FOR I=0 TO 23
3780 RANDOMIZE USR 55500
3790 PAUSE 4
3800 NEXT I
3810 REM
3820 REM *****
3830 REM * SCROLL HACIA ABAJO *
3840 REM *****
3850 REM

```

```

3860 POKE 23300.4 : REM SCROLL HACIA ABAJO
3870 POKE 23301.1 : REM LA PARTE INFERIOR (3)
3880 REM
3890 PRINT AT 0,0:
3900 GO SUB 6000
3910 FOR I=0 TO 7
3920     RANDOMIZE USR 55500
3930     PAUSE 4
3940 NEXT I
3950 REM
3960 POKE 23301.2 : REM LAS DOS PARTES INFERIORES (2 Y 3)
3970 REM
3980 PRINT AT 0,0:
3990 GO SUB 6000
4000 FOR I=0 TO 15
4010     RANDOMIZE USR 55500
4020     PAUSE 4
4030 NEXT I
4040 REM
4050 POKE 23301.3 : REM TODA LA PANTALLA (1,2 Y 3)
4060 REM
4070 PRINT AT 0,0:
4080 GO SUB 6000
4090 FOR I=0 TO 23
4100     RANDOMIZE USR 55500
4110     PAUSE 4
4120 NEXT I
4130 GO SUB 2000
4999 STOP
5000 REM
5010 REM *****
5020 REM * *
5030 REM * ERROR EN LAS LINEAS DATA *
5040 REM * *
5050 REM *****
5060 REM
5070 CLS
5080 PRINT "HA HABIDO UN ERROR EN LAS DATA"
5090 PRINT : PRINT
5100 PRINT "EL ERROR ESTA EN LA LINEA ":LIN
5110 PRINT : PRINT
5120 PRINT "REPASA DICHA LINEA."
5130 PRINT : PRINT
5140 LIST LIN
5150 STOP
6000 REM
6010 REM *****
6020 REM * DIBUJO *
6030 REM *****
6040 REM
6050 FOR I=0 TO 7
6060     PRINT "      ZONA NUMERO UNO"
6070 NEXT I
6080 FOR I=0 TO 7
6090     PRINT "      ZONA NUMERO DOS"
6100 NEXT I
6110 FOR I=0 TO 5
6120     PRINT "      ZONA NUMERO TRES"
6130 NEXT I
6140 PRINT #0:AT 0,0:"      ZONA NUMERO TRES      ZONA NUMERO TRES":
6150 RETURN
9000 REM
9010 REM *****
9020 REM * LINEAS DE DATA *
9030 REM *****
9040 REM
9050 DATA 58,4,91,254,1,40,13,254,2,40,98,254,3,40,90

```

TRUCOS Y RUTINAS BASICAS

```

9055 DATA 1242
9080 DATA 254,4,40,88,201,58,5,91,254,1,40,21,254,2,40
9085 DATA 1353
9070 DATA 24,254,3,40,27,254,4,40,30,254,5,40,33,254,6
9075 DATA 1268
9080 DATA 40,36,201,33,0,64,62,64,24,33,33,0,72,62,64
9085 DATA 788
9090 DATA 24,26,33,0,80,62,64,64,24,19,33,0,64,62,128
9095 DATA 683
9100 DATA 24,12,33,0,72,62,128,24,5,33,0,64,62,192,22
9105 DATA 733
9110 DATA 0,6,31,35,94,43,115,35,16,249,114,35,61,32,242
9115 DATA 1108
9120 DATA 201,24,89,24,85,58,5,91,254,1,40,21,254,2,40
9125 DATA 1189
9130 DATA 24,254,3,40,27,254,4,40,30,254,5,40,33,254,6
9135 DATA 1268
9140 DATA 40,36,201,33,1,72,62,64,24,33,33,255,79,62,64
9145 DATA 1059
9150 DATA 24,26,33,255,87,62,64,24,19,33,255,79,62,128,24
9155 DATA 1175
9160 DATA 12,33,255,87,62,128,24,5,33,255,87,62,192,22,0
9165 DATA 1257
9170 DATA 6,31,43,94,35,115,43,16,249,114,43,61,32,242,201
9175 DATA 1325
9180 DATA 24,88,33,0,0,17,8,0,58,5,91,71,25,16,253
9185 DATA 689
9190 DATA 43,125,50,6,91,33,0,64,17,32,64,229,213,58,6
9195 DATA 1031
9200 DATA 91,79,6,32,26,119,121,230,7,254,1,32,2,151,18
9205 DATA 1169
9210 DATA 35,19,16,241,13,40,19,121,230,7,254,0,40,22,254
9215 DATA 1311
9220 DATA 7,32,225,213,17,0,7,25,209,24,217,209,225,20,36
9225 DATA 1466
9230 DATA 124,254,72,32,202,201,229,33,0,7,25,235,225,24,198
9235 DATA 1861
9240 DATA 33,0,0,17,8,0,58,5,91,71,25,16,253,45,125
9245 DATA 747
9250 DATA 50,6,91,33,255,87,17,223,87,229,213,58,6,91,79
9255 DATA 1525
9260 DATA 6,32,26,119,121,230,7,254,1,32,2,151,18,43,27
9265 DATA 1069
9270 DATA 16,241,13,40,21,121,230,7,254,0,40,24,254,7,32
9275 DATA 1300
9280 DATA 225,213,17,0,7,167,237,82,209,24,215,209,225,21,37
9285 DATA 1888
9290 DATA 124,254,79,200,24,199,229,33,0,7,235,167,237,62,235
9295 DATA 2105
9300 DATA 225,24,193,0,0,0,0,0,0,0,0,0,0,0,0
9305 DATA 442

```

El funcionamiento de la rutina es muy sencillo. En la dirección de memoria 23300 tenemos que poner un número entre 1 y 4 para especificar la dirección en la que nos queremos mover. La siguiente tabla nos da la relación:

```

POKE 23300,1 → SCROLL a la izquierda
POKE 23300,2 → SCROLL a la derecha
POKE 23300,3 → SCROLL hacia arriba
POKE 23300,4 → SCROLL hacia abajo

```

Cualquier otro valor no tendrá ningún efecto.

En la posición 23301 pondremos un número entre 1 y 23. Dicho número tiene distinta función, según estemos realizando un SCROLL vertical u horizontal. En el caso de que el SCROLL sea vertical (arriba o abajo), con este número indicamos el número de líneas que queremos que se muevan. Si el SCROLL es horizontal (derecha o izquierda), dicho número le dice al programa qué parte de la memoria es la que se ha de mover.

La pantalla del SPECTRUM está dividida en tres zonas distintas. La primera va des-

de la línea 0 a la 7. La segunda desde la 8 a la 15 y la tercera desde la 16 a la 23. Si numeramos cada una de estas zonas, esta rutina de SCROLL de caracteres nos permite mover cada una de estas zonas por separado o por grupos.

Por ello, según el número que coloquemos en la posición 23301, el SCROLL afectará a una parte de la pantalla o a otra. La tabla siguiente nos muestra la relación de dichos valores con las zonas de la pantalla:

POKE 23301,1 → SCROLL de la zona 1
 POKE 23301,2 → SCROLL de la zona 2
 POKE 23301,3 → SCROLL de la zona 3
 POKE 23301,4 → SCROLL de las zonas 1 y 2
 POKE 23301,5 → SCROLL de las zonas 2 y 3
 POKE 23301,6 → SCROLL de toda la pantalla

Para ejecutar la rutina sólo hay que llamar a la dirección de memoria 55500 mediante una instrucción USR. Por ejemplo, si en nuestro juego no utilizamos números aleatorios, podemos llamarla con:

RANDOMIZE USR 55500

En el caso de que sí utilicemos números aleatorios, podemos llamarla con la siguiente instrucción:

LET ZZZ=USR 555000

De todas maneras, como la rutina es totalmente reubicable, el usuario puede colocarla en la parte de la memoria que más le convenga.

A continuación, y para aquéllos que estén interesados en el programa fuente de esta rutina, aparece el listado del programa original en ensamblador del Z80.

```

1000 ;*****
1010 ;*
1020 ;*  RUTINAS DE SCROLL *
1030 ;*  PARA SPECTRUM   *
1040 ;*
1050 ;*****
1060 ;
1070 ;*****
1080 ;*
1090 ;*  SCROLL DE CARACTERES *
1100 ;*
1110 ;*****
1120 ;
1130 ;
1140 ;*****
1150 ;*
1160 ;* (C) EDICIONES SIGLO CULTURAL. *
1170 ;*
1180 ;* (C) 1987! *
1190 ;*
1200 ;*****
1210 ;
1220 ;
1230 ;
1240 ;=====
1250 ; LA DIRECCION EN QUE DESEAMOS QUE SE REALIZE
1260 ; EL SCROLL A DE IR ALMACENADA EN LA POSICION
1270 ; DE MEMORIA 23300 SEGUN LA SIGUIENTE TABLA:
1280 ;
1290 ;      1 - SCROLL IZQUIERDA
1300 ;      2 - SCROLL DERECHA
1310 ;      3 - SCROLL ARRIBA
1320 ;      4 - SCROLL ABAJO
1330 ;=====
1340 ;
1350 ;
1360 DIR      EQU      23300
1370 ;
1380 ;

```

TRUCOS Y RUTINAS BASICAS

```

1390 ;=====
1400 ; EL NUMERO DE LINEAS QUE QUEREMOS MOVER SE
1410 ;ALMACENARA EN LA POSICION 23301.
1420 ;=====
1430 ;
1440 ;
1450 NUM      EQU      23301
1460 ;
1470 ;
1480 ;=====
1490 ; LA DIRECCION DONDE SE ALMACENARA LA RUTINA
1500 ;PUEDE SER VARIADA POR EL USUARIO SEGUN SU VO
1510 ;LUNTAD. DE MOMENTO APAREDE EN LA 55500.
1520 ;=====
1530 ;
1540 ;
1550      ORG      55500
1560 ;
1570 ;
1580 ;*****
1590 ;*
1600 ;*  P R O G R A M A  P R I N C I P A L  *
1610 ;*
1620 ;*****
1630 ;
1640 ;
1650      LD      A,(DIR)
1660 ;
1670      CP      1
1680      JR      Z,IZQ
1690 ;
1700      CP      2
1710      JR      Z,DER
1720 ;
1730      CP      3
1740      JR      Z,ARR
1750 ;
1760      CP      4
1770      JR      Z,ABB
1780 ;
1790      RET
1800 ;
1810 ;
1820 ;*****
1830 ;*
1840 ;* SCROLL A LA IZQUIERDA *
1850 ;*
1860 ;*****
1870 ;
1880 ;
1890 IZQ      LD      A,(NUM)
1900 ;
1910      CP      1
1920      JR      Z,IZ-1
1930 ;
1940      CP      2
1950      JR      Z,IZ-2
1960 ;
1970      CP      3
1980      JR      Z,IZ-3
1990 ;
2000      CP      4
2010      JR      Z,IZ-12
2020 ;
2030      CP      5
2040      JR      Z,IZ-23

```

```

2050 ;
2060 CP 6
2070 JR Z, IZ-123
2080 ;
2090 RET
2100 ;
2110 ;
2120 IZ-1 LD HL, 16384
2130 LD A, 64
2140 JR IZ2
2150 ;
2160 IZ-2 LD HL, 18432
2170 LD A, 64
2180 JR IZ2
2190 ;
2200 IZ-3 LD HL, 20480
2210 LD A, 64
2220 JR IZ2
2230 ;
2240 IZ-12 LD HL, 16384
2250 LD A, 128
2260 JR IZ2
2270 ;
2280 IZ-23 LD HL, 18432
2290 LD A, 128
2300 JR IZ2
2310 ;
2320 IZ-123 LD HL, 16384
2330 LD A, 192
2340 ;
2350 ;
2360 IZ2 LD D, 0
2370 ;
2380 LIN-IZ LD B, 31
2390 ;
2400 CHR-IZ INC HL
2410 LD E, (HL)
2420 DEC HL
2430 LD (HL), E
2440 INC HL
2450 DJNZ CHR-IZ
2460 ;
2470 LD (HL), D
2480 INC HL
2490 DEC A
2500 JR NZ, LIN-IZ
2510 ;
2520 RET
2530 ;
2540 ;
2550 ;*** PRIMER GRAN SALTO ***
2560 ;
2570 ;
2580 ARR JR ARR2
2590 ;
2600 ABB JR ABB2
2610 ;
2620 ;
2630 ;*****
2640 ;* *
2650 ;* SCROLL A LA DERECHA *
2660 ;* *
2670 ;*****
2680 ;
2690 ;
2700 DER LD A, (NUM)

```

TRUCOS Y RUTINAS BASICAS

```

2710 ;
2720 CP 1
2730 JR Z,DE-1
2740 ;
2750 CP 2
2760 JR Z,DE-2
2770 ;
2780 CP 3
2790 JR Z,DE-3
2800 ;
2810 CP 4
2820 JR Z,DE-12
2830 ;
2840 CP 5
2850 JR Z,DE-23
2860 ;
2870 CP 6
2880 JR Z,DE-123
2890 ;
2900 RET
2910 ;
2920 ;
2930 DE-1 LD HL,18433
2940 LD A,64
2950 JR DE2
2960 ;
2970 DE-2 LD HL,20479
2980 LD A,84
2990 JR DE2
3000 ;
3010 DE-3 LD HL,22527
3020 LD A,64
3030 JR DE2
3040 ;
3050 DE-12 LD HL,20479
3060 LD A,128
3070 JR DE2
3080 ;
3090 DE-23 LD HL,22527
3100 LD A,128
3110 JR DE2
3120 ;
3130 DE-123 LD HL,22527
3140 LD A,192
3150 ;
3160 ;
3170 DE2 LD D,0
3180 ;
3190 LIN-DE LD B,31
3200 ;
3210 CHR-DE DEC HL
3220 LD E,(HL)
3230 INC CL
3240 LD (HL),E
3250 DEC HL
3260 DJNZ CHR-DE
3270 ;
3280 LD (HL),D
3290 DEC HL
3300 DEC A
3310 JR NZ,LIN-DE
3320 ;
3330 RET
3340 ;
3350 ;
3360 ;*** SEGUNDO GRAN SALTO ***

```

```

3370 ;
3380 ;
3390 ABB2      JR      ABB3
3400 ;
3410 ;
3420 ;*****
3430 ;*          *
3440 ;* SCROLL HACIA ARRIBA *
3450 ;*          *
3460 ;*****
3470 ;
3480 ;
3490 ARR2      LD      HL, 0
3500          LD      DE, 8
3510          LD      A, (NUM)
3520          LD      B, A
3530 ;
3540 SUM-AR    ADD     HL, DE
3550          DJNZ    SUM-AR
3560 ;
3570          DEC     HL
3580          LD      A, L
3590          LD      (23302), A
3600 ;
3610          LD      HL, 16384
3620          LD      DE, 16416
3630 ;
3640 PON-AR    PUSH    HL
3650          PUSH    DE
3660          LD      A, (23302)
3670          LD      C, A
3680 ;
3690 LIN-AR    LD      B, 32
3700 ;
3710 MOV-AR    LD      A, (DE)
3720          LD      (HL), A
3730          LD      A, C
3740          AND     7
3750          CP      1
3760          JR      NZ, CHR-AR
3770 ;
3780          SUB     A
3790          LD      (DE), A
3800 ;
3810 CHR-AR    INC     HL
3820          INC     DE
3830          DJNZ    MOV-AR
3840 ;
3850          DEC     C
3860          JR      Z, FIN-AR
3870 ;
3880          LD      A, C
3890          AND     7
3900          CP      0
3910          JR      Z, PA-AR
3920 ;
3930          CP      7
3940          JR      NZ, LIN-AR
3950 ;
3960          PUSH    DE
3970          LD      DE, 1792
3980          ADD     HL, DE
3990          JR      LIN-AR
4000 ;
4010 FIN-AR    POP     DE
4020          POP     HL

```

TRUCOS Y RUTINAS BASICAS

```

4030      INC      D
4040      INC      H
4050      LD       A,H
4060      CP       72
4070      JR       NZ,PON-AR
4080 ;
4090      RET
4100 ;
4110 PA-AR      PUSH    HL
4120      LD       HL,1792
4130      ADD      HL,DE
4140      EX       DE,HL
4150      POP      HL
4160      JR       LIN-AR
4170 ;
4180 ;
4190 ;*****
4200 ;*          *
4210 ;* SCROLL HACIA ABAJO *
4220 ;*          *
4230 ;*****
4240 ;
4250 ;
4260 ABB3      LD      HL,0
4270      LD      DE,8
4280      LD      A,(NUM)
4290      LD      B,A
4300 ;
4310 SUM-AB     ADD     HL,DE
4320      DJNZ    SUM-AB
4330 ;
4340      DEC     L
4350      LD      A,L
4360      LD      (23302),A
4370      LD      HL,22527
4380      LD      DE,22495
4390 ;
4400 PON-AB     PUSH    HL
4410      PUSH    DE
4420      LD      A,(23302)
4430      LD      C,A
4440 ;
4450 LIN-AB     LD      B,32
4460 ;
4470 MOV-AB     LD      A,(DE)
4480      LD      (HL),A
4490      LD      A,C
4500      AND     7
4510      CP     1
4520      JR     NZ,CHR-AB
4530 ;
4540      SUB     A
4550      LD      (DE),A
4560 ;
4570 CHR-AB     DEC     HL
4580      DEC     DE
4590      DJNZ    MOV-AB
4600 ;
4610      DEC     C
4620      JR     Z,FIN-AB
4630 ;
4640      LD      A,C
4650      AND     7
4660      CP     0
4670      JR     Z,PO-AB
4680 ;

```

4690		CP	7
4700		JR	NZ, LIN-AB
4710 ;			
4720		PUSH	DE
4730		LD	DE, 1792
4740		AND	A
4750		SBC	HL, DE
4760		POP	DE
4770		JR	LIN-AB
4780 ;			
4790 FIN-AB		POP	DE
4800		POP	HL
4810		DEC	D
4820		DEC	H
4830		LD	A, H
4840		CP	79
4850		RET	Z
4860 ;			
4870		JR	PON-AB
4880 ;			
4890 PO-AB		PUSH	HL
4900		LD	HL, 1792
4910		EX	DE, HL
4920		AND	A
4930		SBC	HL, DE
4940		EX	DE, HL
4950		POP	HL
4960		JR	LIN-AB

EL TALLER DEL HARDWARE

■ Mesa digitalizadora



IN duda, se nos han presentado situaciones en las que hubiera sido conveniente disponer de un medio de comunicación con el ordenador de forma gráfica, para introducir los resultados de un experimento físico, la curva contenida en un libro o un dibujo trazado a mano alzada. En cualquiera de los casos el ideal sería que el ordenador «viera» el dibujo y que lo convirtiera en datos manipulables fácilmente por programa.

Existen medios para comunicar al ordenador datos de forma gráfica y de uno de ellos ya se ha propuesto un prototipo, al describir los tipos de sensores resistivos, pero hay otras posibilidades de mayor resolución y que es conveniente conocer, aunque resulten de coste algo más elevado y con una dificultad de construcción mayor. El circuito propuesto en un tomo anterior se basaba en resistencias variables y en su utilización para medir desplazamientos.

El montaje que en esta ocasión vamos a tratar es una mesa digitalizadora de tipo electromagnético, conectable a nuestros ordenadores personales a través de la tarjeta de ampliación de puertos de entrada-salida. Más adelante mostraremos las posibilidades de las cámaras de vídeo como elementos utilizables para la captación directa de imágenes para nuestro ordenador.

■ Principio de funcionamiento

El método es bastante sofisticado y permite obtener mayor precisión y resolución que el basado en resistencias variables, está basado en los sensores de tipo inductivo. Un transformador de núcleo de aire en el cual los bobinados puedan moverse uno respecto del otro es una explicación simplificada del fenómeno en el que se fundamentan.

Un transformador de corriente alterna es un dispositivo eléctrico que permite la transferencia de energía de un circuito a otro, sin conexión metálica entre los dos circuitos. La corriente eléctrica se transforma en campo magnético en el circuito primario y se reconvierte a corriente eléctrica en el secundario. En los transformadores ordinarios los circuitos primario y secundario se suelen hacer arrollados sobre un núcleo de hierro, para que el grado de acoplamiento sea el mayor posible y el campo magnético disperso sea el menor posible. En los circuitos de alta frecuencia, como, por ejemplo, para los aparatos de radio se emplean materiales magnéticos basados en hierro, pero que posean menores pérdidas a las corrientes originadas en el interior. Cuando la frecuencia es muy alta, el núcleo empleado es solamente de aire. Si mantenemos uno de los bobinados fijo y el otro móvil, podemos observar que, para una corriente alterna introducida en el primario, aparece en el secundario una corriente proporcional a la primaria y a la proximidad entre los bobinados. Hemos «descubierto» un tipo de transductor de posición a señal eléctrica proporcional.

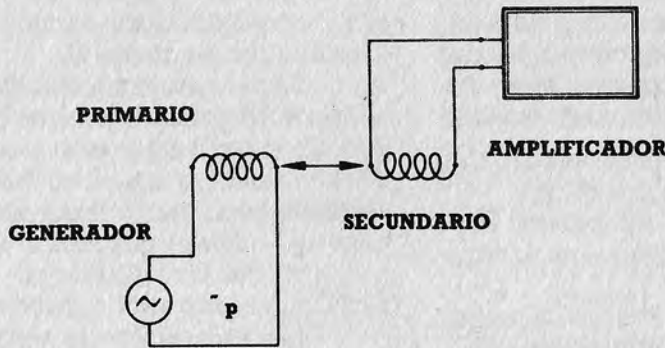


Fig. 1. Transformador de núcleo de aire y secundario móvil.

Si además de desear que sea medible la posición es necesario que se haga en dos dimensiones, deberemos «planchar» el primario, para que sobre él pueda fácilmente desplazarse el secundario móvil.

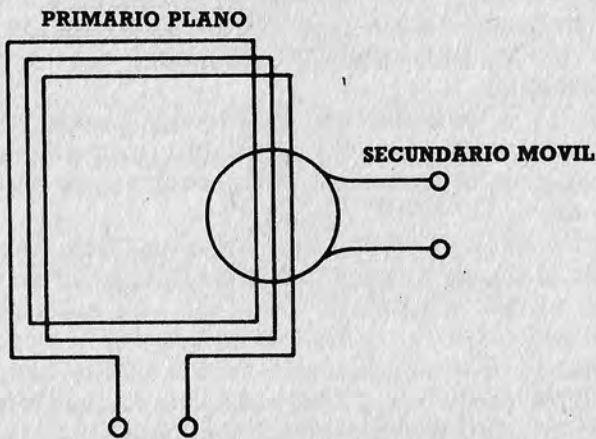


Fig. 2. Primario plano y secundario móvil.

Para que los desplazamientos originen una magnitud medible por el procedimiento descrito es necesario que el bobinado secundario se encuentre en la proximidad de una espira del primario, para que la señal sea suficiente y no se encuentre enmascarada por el ruido. Por ello montaremos un circuito que bá-

sicamente es un transformador de núcleo de aire con el circuito primario plano y en el que pueda variarse la geometría del circuito realmente activo del primario. Si el secundario se mueve dentro del área creada por el primario y en su mismo plano, la señal recibida será prácticamente independiente de su posición. Si el secundario se encuentra en el borde del bobinado del primario, entonces sí que aparecen variaciones de amplitud para diferentes posiciones de la bobina sensora. Por esta razón deberemos hacer que el circuito primario cambie de forma para que haya siempre un cambio en las proximidades de la bobina sensora.

La forma de onda de la señal empleada en el transformador de núcleo de aire es muy importante, para que sea fácilmente detectable con los circuitos de captación de la señal.

■ Diagrama de bloques

El esquema de bloques muestra los elementos principales:

— Oscilador. Define la frecuencia de referencia de generación de los pulsos. Es

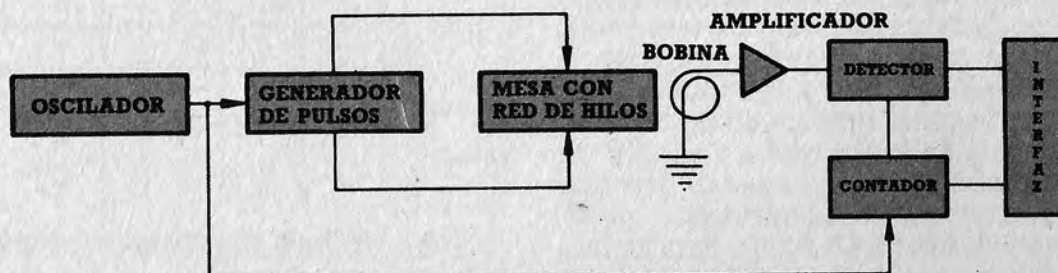


Fig. 3. Diagrama de bloques.

EL TALLER DEL HARDWARE

conveniente que sea de frecuencia ajustable, para poder adaptarse a las características del circuito de entrada de señal, pero una vez seleccionada conviene mantenerla muy estable.

- Contador de posición. Determina la cuenta que define la posición del cursor.

- Circuito generador de pulsos. Produce secuencialmente los pulsos necesarios en cada hilo de la rejilla.

- Bobina captadora. Recibe la señal producida por los hilos. Para un montaje práctico debe ser de 50 espiras sobre un aro circular de 5 centímetros de diámetro. Para que la señal captada resulte apropiada deber formar circuito casi resonante con el condensador de entrada al amplificador.

- Amplificador. Eleva el nivel de la señal captada por la bobina para hacerla utilizable en la detección.

- Detector síncrono. Toma muestras de la señal, en sincronismo con el generador de pulsos para captar la envolvente de la señal de entrada. La etapa de filtro se utiliza para rechazar las frecuencias superiores, donde normalmente se encontrarán los ruidos. El comparador determina el paso por cero de la señal, que es el instante utilizado para determinar el defase de la señal con respecto a un origen arbitrario, tomado como referencia.

- Puede añadirse un indicador de proximidad. Sirve para determinar si el nivel de la señal de entrada es suficiente para efectuar la medida.

- Circuito de interfaz con el ordenador. Presenta el estado y el valor del contador al final de la detección.

■ Circuito primario del transformador

El circuito primario, necesario para nuestros propósitos, consta de una tabla o, mejor todavía, dos láminas de plástico transparente, con una rejilla de hilos uniformemente espaciados. Cada uno de los hilos se conecta, por un lado, al circuito de señal y, por otro, al multiplexor, como puede verse en la figura. Es un primario de transformador un poco «raro», pues solamente tiene una espira y además de posición variable. El ejemplo que se describe permite la construcción de una mesa digitalizadora simplificada de 8 x 8 hilos. Para un funcionamiento correcto los hilos deben estar uniformemente espaciados, aunque la distancia exacta no es importante para cualquier reso-

lución deseada; ésta es más dependiente de los circuitos electrónicos.

El proceso completo de funcionamiento consiste en generar un tren de pulsos que se aplican sucesivamente a cada uno de los hilos de la rejilla, primero en una dirección, por ejemplo, horizontal y luego en la otra, vertical. Esta es la forma particular de utilización del primario del transformador: en cada impulso varía el circuito real utilizado.

La generación de pulsos se puede hacer simplemente por aplicación de la salida de un nivel de 5 voltios producido por un conmutador de transistores, mientras se activa la salida de un selector/multiplexor que pasa al estado de conducción. La intensidad en los hilos de la rejilla se limita mediante las resistencias indicadas en el esquema del circuito. Los pulsos aplicados a los hilos generan un campo electromagnético que puede ser captado mediante una bobina colocada en su proximidad y que hace las veces del secundario del transformador.

El hilo utilizado en la rejilla puede ser de dos o tres décimas de milímetro de diámetro, pues la corriente media recorrida es muy baja.

El funcionamiento global es, pues, similar al de un transformador de núcleo de aire en el cual el primario es de una sola espira y el secundario es la bobina que se utiliza como cursor. El procedimiento básico puede cambiarse por el complementario, en el que es la bobina la que emite el pulso y son los hilos de la rejilla los que captan la radiación.

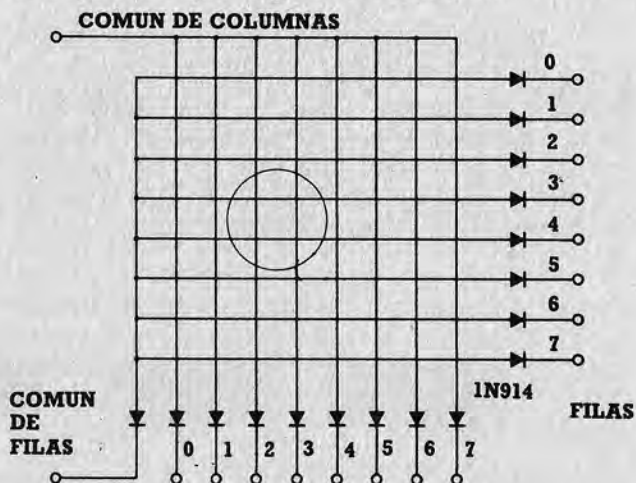


Fig. 4. Rejilla del primario y circuito de conexión.

Para el procedimiento básico, en el que son los hilos los que emiten y la bobina la que capta la radiación, puede verse cómo hay una

magnitud, fácilmente medible, que varía con la posición. Si nos ceñimos a la exploración horizontal y para el caso de que la bobina se encuentre en una posición intermedia, como la que indica la figura. Al efectuarse el barrido completo de izquierda a derecha, la bobina captadora recibirá una señal proporcional a la corriente en el hilo activo. Si la bobina no se encuentra sobre el hilo, la señal recibida será casi independiente de la posición relativa al

hilo, pues lo que afecta a las bobinas acopladas es el área mutua. Por lo que respecta a la fase de la señal recibida con relación a la del primario depende de si la bobina se encuentra a la derecha o a la izquierda del hilo considerado, es decir, en un sentido u otro del flujo. Si se encuentra dentro, la señal estará en fase, y si se encuentra fuera, en contrafase. Habrá, pues, una diferencia de 180 grados en cada caso.

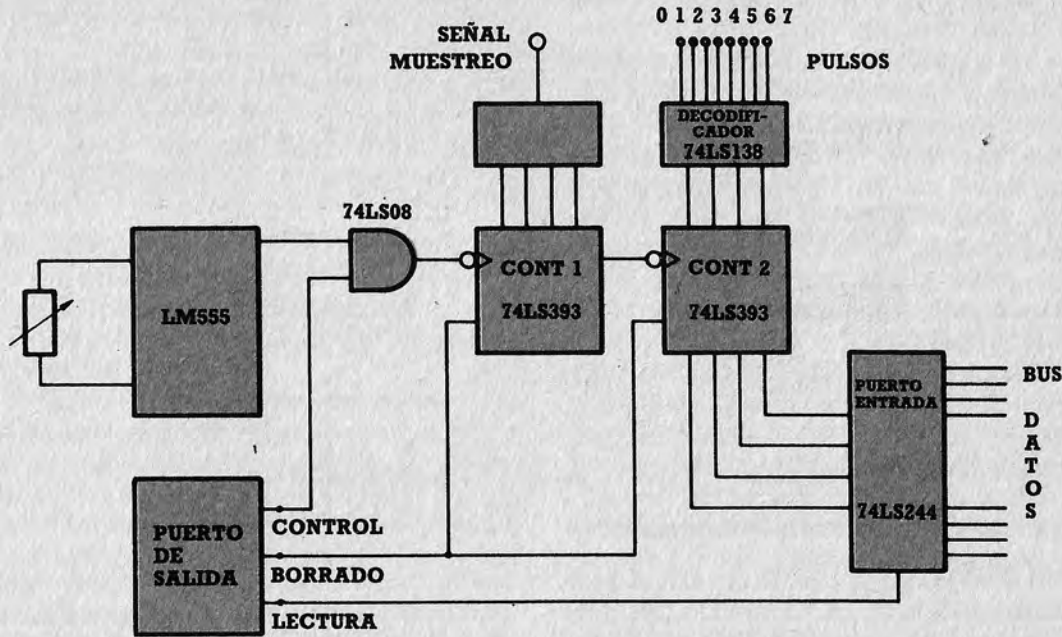


Fig. 5. Circuito generador de pulsos para el primario.

Para una posición en la que el hilo emisor se encuentre dentro del área cubierta por la bobina, habrá una composición que, como puede comprenderse, será destructiva, por lo que la amplitud de la señal será menor que en las posiciones con cubrimiento completo. Para

posiciones intermedias irá decreciendo a medida que el hilo toma posiciones centradas con relación a la bobina. Para el caso en que el hilo esté sobre el centro de la bobina sensora, la señal recibida correspondiente al hilo que emite será nula.

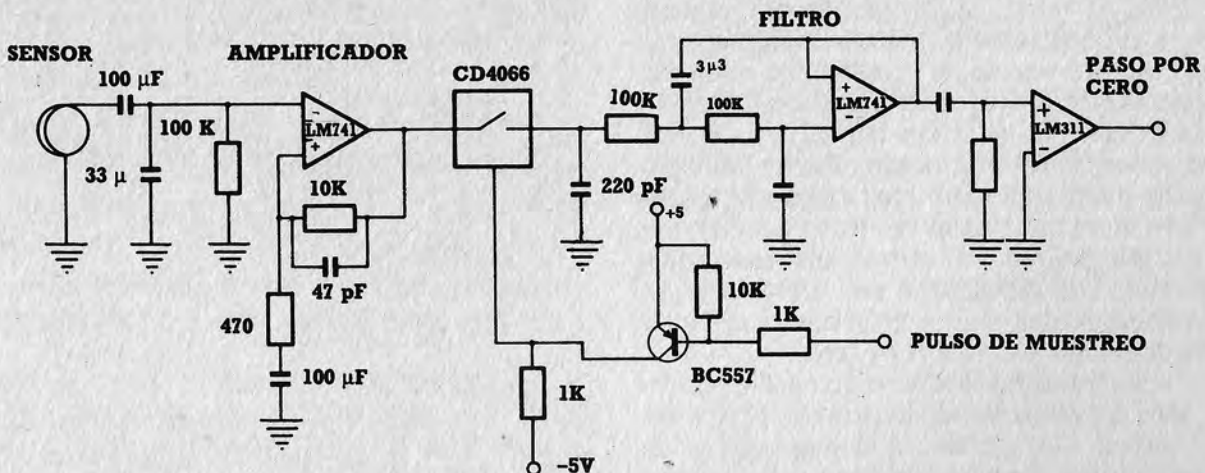


Fig. 6. Circuito de captación y tratamiento de la señal analógica.

EL TALLER DEL HARDWARE

La variación de la señal con la posición permite obtener mediante detección y filtrado la componente principal de la señal compuesta de la captación de las señales emitidas por todos los hilos de la rejilla (A). Mediante detección síncrona puede recuperarse la en-

volvente de las amplitudes instantáneas de cada pulso (B). Esta composición es necesario tenerla en cuenta, pues el efecto de un pulso sobre un sistema tiene duración finita. Esta señal filtrada presenta la forma (C) indicada en la figura 7.

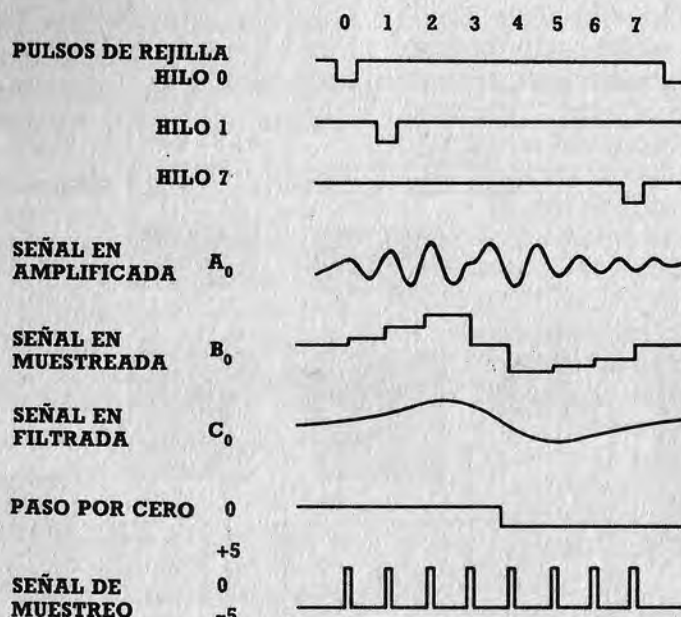


Fig. 7. Señales en el circuito de lectura en diferentes posiciones.

Para convertir esta señal al valor de posición correspondiente, es necesario detectar el paso por cero de la amplitud de dicha señal, pues ese punto indica el momento en el que hay un cambio de fase. El contador utilizado para medir la posición debe ser borrado antes de cada comienzo de barrido y activado a la vez que comienza la generación de pulsos sobre los hilos. El contador se parará al detectarse el paso por cero de la envolvente de amplitud. El valor que indique en ese momento será proporcional al número de pulsos generado y con variación lineal entre cada dos pulsos. El circuito de control hará que después de la lectura del valor del contador se permita el comienzo de otra lectura. Puede seleccionarse el modo de lectura, para que se haga de manera continua o cada vez que se active uno de los pulsadores de los que conviene dotar al cursor. Los pulsadores se conectan como entradas binarias, de forma similar a como se han conectado en otros proyectos.

Es conveniente añadir una señal adicional para la indicación de proximidad de la bobina cursor a la tableta. Si la amplitud no alcanza un nivel preestablecido, se indicará una condición de error y no se darán por válidas

las lecturas. Para determinar si la amplitud es correcta se instalará un comparador a la salida del amplificador, estableciendo el nivel de comparación con una resistencia variable conectada a la alimentación positiva.

Programa de lectura para el IBM-PC

```

10 REM TABLA ELE
20 REM DIGITALIZADOR MEDIANTE
   REJILLA Y BOBINA
30 REM A TRAVES DE TARJETA DE
   AMPLIACION DE PUERTOS
40 REM SE ACTIVA LA LECTURA POR
   PULSADOR 0
50 ENTRADA=&H201: ESTADO=&H203:
   CONTROL=&H201
60 KEY OFF: CLS
70 ON STRIG(0) GOSUB 150
80 STRIG(0) ON
90 SCREEN 2: CLS
100 PRINT "LECTURA DE COORDENA-
   DAS"
110 FOR I=1 TO 10000
120 REM PROGRAMA DE APLICA-
   CION, EN BUCLE
130 NEXT I
140 END
  
```

```

150 REM ACTIVACION Y LECTURA DE
COORDENADAS
160 OUT CONTROL,ACTX
170 M=INP(ESTADO)
175 IF M=1 THEN 170
180 X=INP(ENTRADA)
200 OUT CONTROL,ACTY
210 M=INP(ESTADO): REM LEE SI CON-
TADOR PARADO
220 IF M=1 THEN 210
230 Y=INP(ENTRADA)
240 PSET (X,Y),1: REM MARCA EL PUN-
TO
250 LOCATE 25,1: REM LISTA COOR-
DENADAS, EN TEST
260 PRINT "I=";I," X=";X," Y=";Y;
270 RETURN

```

El programa incluye presentación en pantalla de las coordenadas leídas en el momento de la pulsación, utilizable en calibración y pruebas; en una aplicación normal podría suprimirse.

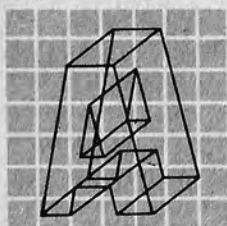
El circuito que se propone como ejemplo puede ampliarse en tamaño o en el número de líneas utilizadas para explorar la mesa, simplemente aumentando el número de etapas del contador y, por consiguiente, el de circuitos de multiplexión. El esquema de circuito

de recepción y tratamiento de la señal de la bobina puede ser el mismo, cualquiera que sea el número de líneas.

Los equipos comerciales utilizan procedimientos similares al descrito, ofreciendo resoluciones del orden de una décima de milímetro. La forma más cómoda y eficiente de control del barrido se realiza mediante un microcontrolador monochip, realizándose la comunicación a través de puerto asíncrono serie, con objeto de hacerlo conectable a cualquier ordenador. Suelen ser programables en cuanto a número de medidas por segundo, incremento de separación mínimo para realizar una medida, formato de las coordenadas, capacidad de enviar identificación de un punto, hasta 12 teclas de función, cursor tipo lápiz o espira con cruz y posibilidad de indicar tercera coordenada. Para esta posibilidad de medida se sustituye la espira plana por una bobina cilíndrica de cierta longitud, pudiendo entonces determinar con bastante aproximación no sólo las coordenadas de la posición del centro de la bobina, sino además su inclinación con respecto a los ejes. Esta tercera coordenada puede emplearse para indicar alguna otra característica del punto digitalizado, como, por ejemplo, la pendiente de una superficie o el color de representación.

NATURALEZA Y TECNOLOGIA: GEOLOGIA

■ Test sobre geología (minerales)



continuación te ofrecemos un programa con el que vas a tener la oportunidad de aprender o recordar distintos nombres de minerales y las características de cada uno a través de un entretenido test. El programa tiene

la misma estructura que otros que ya se han publicado en esta misma sección sobre diferentes temas. Al darle la orden de ejecución, escribe en pantalla una serie de características correspondientes a un mineral y, a continuación, escribirá la primera y la última letra del nombre de éste, así como el número de letras que faltan. Seguidamente tú deberás ir tecleando las letras que creas que faltan para completar el nombre. Cuando teclees una letra que está en el nombre, el programa te la colocará en su lugar adecuado y deberás continuar hasta que termines el nombre. En la siguiente figura podemos observar una imagen de cómo tendríamos la pantalla en un momento determinado de la ejecución mientras estamos intentando adivinar el nombre.

El sistema es el mismo que el del popular juego del «Ahorcado».

Silicato de aluminio empleado para la fabricación de porcelanas.

C A L T A

Teclea las letras que creas que faltan en la palabra (mayúsculas).

■ Explicación del programa

El esquema general del programa es el mismo que el de los otros publicados en números anteriores basados en el mismo juego, variando en este caso, evidentemente, los datos correspondientes a las definiciones (en nuestro caso descripción de características de minerales) y los correspondientes a los datos de los nombres de los minerales.

```
10 REM *****
20 REM **
30 REM **          TEST SOBRE GEOLOGIA (MINERALES)          **
40 REM **
50 REM *****
60 DIM P$(20,11):DIM L$(11)
70 REM LA INSTRUCCION 80 SOLO ES NECESARIO UTILIZARLA PARA MS-DOS (IBM O
  COMPATIBLES)
80 RANDOMIZE TIMER
90 DATA 67,85,65,82,90,79, 0,84,79,80,65,67,73,79, 0,77,65,71,78,69,84,73,84,65
100 DATA 0,80,76,65,84,73,78,79, 0,80,76,65,84,65, 0,80,73,82,73,84,65, 0,79,76
110 DATA 73,71,73,83,84,79, 0,77,69,82,67,85,82,73,79, 0,71,82,65,70,73,84,79,0
120 DATA 71,82,65,78,73,84,79, 0,68,73,65,77,65,78,84,69, 0,67,79,66,82,69, 0,67
```

```

130 DATA 85,80,82,73,84,65, 0,67,73,78,65,66,82,73,79, 0,66,76,69,78,68,65, 0,70
140 DATA 76,85,79,82,73,84,65, 0,84,65,76,67,79, 0,67,65,79,76,73,78,73,84,65,0
150 DATA 66,65,83,65,76,84,79, 0,67,65,76,67,79,80,73,82,73,84,65,0
160 PRINT :PRINT "¡UN MOMENTO!, ESTOY CARGANDO LOS DATOS"
170 FOR I=1 TO 20
180 LET J=1
190 READ A
200 IF A=0 THEN FOR K=J TO 11 :LET P$(I,K)=" ":NEXT K :GOTO 230
210 LET P$(I,J)=CHR$(A)
220 LET J=J+1:GOTO 190
230 NEXT I
240 LET PAL=INT(RND(1)*20)+1
250 FOR J=1 TO 11 :LET L$(J)=" ":NEXT J
260 LET FT=0 :LET CON=0
270 CLS
280 ON PAL GOTO 290,300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,
    450,460,470,480
290 PRINT "ANHIDRIDO SILICICO DE BRILLO VITREO UTILIZADO COMO MATERIAL DE LABORA
TORIO":GOTO 490
300 PRINT "MINERAL UTILIZADO PARA PIEDRAS PRECIOSAS EN SU COLORACION AMARILLA":G
OTO 490
310 PRINT "OXIDO FERROSO FERRICO.DENOMINADA PIEDRA IMAN POR SU PODER DE ATRAER E
L HIERRO":GOTO 490
320 PRINT "METAL PRECIOSO UTILIZADO EN JOYERIA DE COLOR GRIS":GOTO 490
330 PRINT "METAL PRECIOSO DUCTIL Y MALEABLE,BUEN CONDUCTOR DEL CALOR Y LA ELECTR
ICIDAD":GOTO 490
340 PRINT "SULFURO DE HIERRO DE COLOR AMARILLO.LATON EMPLEADO PARA LA OBTENCION
DEL ACIDO SULFURICO":GOTO 490
350 PRINT "SESQUIOXIDO DE HIERRO (FE2 O3).TIENE IMPORTANTES YACIMIENTOS EN VIZCA
YA Y ARAGON (OJOS NEGROS Y TIERRA)":GOTO 490
360 PRINT "UNICO METAL LIQUIDO A TEMPERATURA ORDINARIA QUE SOLIDIFICA A -40 GRAD
OS C.":GOTO 490
370 PRINT "FORMA ALOTROPICA DEL CARBONO DE COLOR NEGRO UNTUOSO EXFOLIABLE EN LAM
INILLAS":GOTO 490
380 PRINT "MINERAL QUE TAMBIEN RECIBE EL NOMBRE DE PIEDRA BERROQUENA":GOTO 490
390 PRINT "CARBONO PURO.EL MAS DURO DE LOS MINERALES.COMPLETAMENTE INATACABLE E
INFUSIBLE":GOTO 490
400 PRINT "METAL DE COLOR NEGRO INTENSO, DUCTIL Y MALEABLE, BUEN CONDUCTOR DE LA
ELECTRI- CIDAD":GOTO 490
410 PRINT "OXIDO DE COBRE DE COLOR ROJO":GOTO 490
420 PRINT "SULFURO DE MERCURIO DE COLOR ROJO FUERTE CUYO YACIMIENTO MAS IMPORTAN
TE DEL MUNDO ESTA EN ALMADEN(CIUDAD REAL)":GOTO 490
430 PRINT "SULFURO DE ZINC. FRAGIL. TIENE DIFERENTES COLORES SEGUN LAS IMPUREZAS
; CON MUCHO HIERRO ES CASI NEGRA":GOTO 490
440 PRINT "FLUORURO CALCICO EMPLEADO PARA LENTES DE MICROSCOPIOS":GOTO 490
450 PRINT "SILICATO MAGNESICO HIDRATADO EMPLEADO PARA POLVOS DE ASEO":GOTO 490
460 PRINT "SILICATO DE ALUMINIO EMPLEADO PARA LA FABRICACION DE PORCELANAS":GOTO
490
470 PRINT "ROCA BASICA DE COLOR OSCURO, PESADA, COMPACTA Y RESISTENTE EMPLEADA P
ARA ADO- QUINADOS":GOTO 490
480 PRINT "SULFURO DE HIERRO Y COBRE DE COLOR AMARILLO VERDOSO. MENA PRINCIPAL D
EL COBRE"
490 LET L$(1)=P$(PAL,1)
500 FOR I=11 TO 2 STEP-1
510 IF P$(PAL,I)<>" " THEN LET L$(I)=P$(PAL,I):LET F=I:GOTO 530
520 NEXT I
530 LOCATE 10,10
540 FOR I=1 TO F
550 PRINT L$(I); " ";
560 NEXT I
570 PRINT:PRINT TAB(12);
580 FOR I=2 TO F-1
590 PRINT"- ";
600 NEXT I
610 REM PARA AMSTRAD Y MSX :620 LOCATE 1,15
620 LOCATE 15,1
630 IF FT THEN GOTO 750
640 LET FT=1
650 PRINT "TECLEA LAS LETRAS QUE CREAS QUE FALTEN EN LA PALABRA(MAYUSCULAS)"
660 LET A$=INKEY$
670 IF A$="" THEN GOTO 660
680 FOR I=2 TO F-1
690 IF P$(PAL,I)=A$ THEN LET L$(I)=P$(PAL,I):LET LT=1
700 IF L$(I)=" " THEN LET FT=0
710 NEXT I
720 IF LT=0 THEN CON=CON+1 :PRINT:PRINT"LO SIENTO,LA LETRA ";A$;" NO ESTA"
730 LET LT=0
740 GOTO 530
750 PRINT "
760 PRINT :PRINT "
770 PRINT ";ENHORABUENA!,LA HAS ACERTADO":PRINT
780 PRINT "HAS TECLEADO ";CON;" LETRAS ERRONEAS AL INTENTAR ACERTAR LA PALABRA"
790 PRINT "QUIERES JUGAR OTRA VEZ? (S/N)"
800 LET A$=INKEY$

```

APRENDER CON EL ORDENADOR

```

810 IF A$="S" THEN GOTO 240
820 IF A$(">"N" THEN GOTO 800
830 REM *****
840 REM ** MODIFICACIONES PARA COMMODORE : **
850 REM ** 270 PRINT CHR$(147) **
860 REM ** 530 FOR I=1 TO 8 **
870 REM ** 532 PRINT **
880 REM ** 534 NEXT I :PRINT TAB(10); **
890 REM ** 620 FOR I=1 TO 4 **
900 REM ** 623 PRINT **
910 REM ** 625 NEXT I **
920 REM ** 660 GET A$ **
930 REM ** 740 GOTO 270 **
940 REM ** 800 GET A$ **
950 REM ** MODIFICACIONES PARA SPECTRUM **
960 REM ** 280 GOTO (PAL*10)+280 **
970 REM ** 530 PRINT AT 10,10 **
980 REM ** 620 PRINT AT 15,1 **
990 REM *****

```

A continuación se ofrece el comentario de éste, que, como ya se ha indicado, no difiere mucho de los realizados en los programas publicados anteriormente y basados en el mismo juego.

El programa comienza con siete instrucciones DATA en las que se introducen los nombres de los minerales. Como se observará a la vista del programa, los datos de estas instrucciones no se parecen lo más mínimo a una lista de nombres, sino que son una lista ordenada de números. Estos números precisamente se corresponden con los códigos internos (códigos ASCII) que las letras tienen dentro de nuestro ordenador. Cada número es una letra. La razón de introducir en los DATA los códigos de las letras en vez de las propias letras directamente es obvia: si introdujésemos ahí los nombres de los minerales, al teclear el programa en nuestro ordenador tendríamos que teclear estos nombres y cuando posteriormente lo ejecutásemos ya sabríamos estos nombres, con lo que el juego perdería su interés. En las instrucciones DATA, al final de los números correspondientes a las letras de cada una de las palabras, se ha colocado un cero para que el programa pueda localizar cuando los lea el final de cada palabra.

El programa guarda las palabras que lee, después de transformar los códigos en letras mediante la función CHR\$ en la tabla P\$ (instrucciones 170-230). Hay veinte nombres de minerales en total y el máximo de letras que pueden tener es once. En aquellos que tienen menos de once letras se rellenan con blancos los espacios sobrantes.

En la instrucción 240 el programa selecciona un número al azar entre 1 y 20, que será el número del mineral que habrá que adivinar. Entonces escribe el texto correspondien-

te a sus características mediante la instrucción ON (línea 280).

El programa guarda en la lista L\$ la palabra con las letras que tú ya has adivinado teniendo blancos en los espacios correspondientes a las letras que aún te quedan por acertar. Así, inicialmente entre las sentencias 490 y 520 guarda en L\$ la primera y última letra de la palabra en sus lugares correspondientes.

Entre las instrucciones 530 y 600 se escriben en la pantalla la letra inicial y final, un guión por cada letra que hay que adivinar, así como todas aquellas letras que haya en L\$, que serán las que hayamos acertado ya cuando se ejecuten estas instrucciones de nuevo.

Cuando introduces una letra (que se guarda en A\$ en la instrucción 660) se comprueba si está, y si es así se guarda en su lugar correspondiente de L\$ (instrucciones 660-710) para luego volver a 530, donde escribe de nuevo L\$ con las letras introducidas.

Significado de las variables de control que utiliza el programa:

LT=0 → Indica que la última letra introducida no está en la palabra.

LT=1 → La última letra introducida sí está en la palabra.

FT=1 → Indica que ya se ha acertado la palabra.

FT= → La palabra aún no se ha acertado.

F → Número de letras de la palabra (río) que se está adivinando en cada momento.

Las variables LT y FT se ponen a 0 y 1, respectivamente, cada vez que empezamos una nueva operación (líneas 640 y 730) y si luego se comprueba que lo que indican estos valores no se cumple, se les cambia de valor.

NOTA: El programa, como ya se ha dicho anteriormente, selecciona de manera

aleatoria el mineral que va a sacar, por lo que puede darse el caso de que a veces se repita uno mismo dos veces casi seguidas, pero, sin embargo, esta técnica, de no seguir un orden preestablecido, aumenta el interés del juego, ya que cuando ejecutes el programa de nuevo en otras ocasiones no sabrás cuáles te van a salir.

MATEMATICAS

■ Resolución de un sistema de n ecuaciones con n incógnitas

Existen numerosos problemas, tanto de matemáticas como de otras materias, que se apoyan en las matemáticas, en los cuales es necesario resolver un sistema de n ecuaciones con n incógnitas.

El programa que se ofrece a continuación en esta sección nos resuelve cualquier sistema de éstos.

El método más sencillo para resolver estos sistemas (que es el que utiliza el programa) es la regla de Cramer, cuyo funcionamiento es el siguiente:

Supongamos que tenemos el siguiente sistema, ya debidamente ordenado:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= t_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= t_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= t_n \end{aligned}$$

Fig. 2.

La fórmula para hallar el valor de las distintas incógnitas X_i es:

$$X_i = \frac{P_i}{C} = \frac{\begin{vmatrix} a_{11} & a_{12} & \dots & t_1 & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & t_2 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & t_n & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1i} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2i} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{ni} & \dots & a_{nn} \end{vmatrix}}$$

Fig. 3.

en donde, como puede observarse, C es el determinante de la matriz formada por los coeficientes de las incógnitas y P_i el determinante de la matriz formada al sustituir en la matriz de los coeficientes la columna i por la de los términos independientes.

Aplicado este método al sistema del siguiente ejemplo nos quedaría:

$$\begin{aligned} 4x_1 + 5x_2 + 3x_3 - 2x_4 &= 7 \\ 8x_1 + 9x_2 + 15x_3 + 4x_4 &= 24 \\ x_1 - x_2 + 3x_3 - x_4 &= 18 \\ 7x_1 - 5x_2 + 4x_3 - 18x_4 &= 31 \end{aligned}$$

$$C = \begin{vmatrix} 4 & 5 & 3 & -2 \\ 8 & 9 & 15 & 4 \\ 1 & -1 & 3 & -1 \\ 7 & -5 & 4 & -18 \end{vmatrix}$$

$$\begin{aligned} x_1 &= \frac{\begin{vmatrix} 7 & 5 & 3 & -2 \\ 24 & 9 & 15 & 4 \\ 18 & -1 & 3 & -1 \\ 31 & -5 & 4 & -18 \end{vmatrix}}{C} & x_3 &= \frac{\begin{vmatrix} 4 & 5 & 7 & -2 \\ 8 & 9 & 24 & 4 \\ 1 & -1 & 18 & -1 \\ 7 & -5 & 31 & -18 \end{vmatrix}}{C} \\ x_2 &= \frac{\begin{vmatrix} 4 & 7 & 3 & -2 \\ 8 & 24 & 15 & 4 \\ 1 & 18 & 3 & -1 \\ 7 & 31 & 4 & -18 \end{vmatrix}}{C} & x_4 &= \frac{\begin{vmatrix} 4 & 5 & 3 & 7 \\ 8 & 9 & 15 & 24 \\ 1 & -1 & 3 & 18 \\ 7 & -5 & 4 & 31 \end{vmatrix}}{C} \end{aligned}$$

Fig. 4.

■ Comentario del programa

El programa en cuestión es el siguiente:

```

10 REM *****
20 REM **
30 REM ** PROGRAMA QUE RESUELVE UN SISTEMA DE N ECUACIONES CON N INCOGNITAS **
40 REM **
50 REM *****

```

APRENDER CON EL ORDENADOR

```

60 CLS
70 INPUT "¿CUANTAS INCOGNITAS (ECUACIONES) TIENE EL SISTEMA";N
80 DIM C(N,N):DIM T(N):DIM D(N,N):DIM P(N,N,N):DIM S(N*2)
90 CLS
100 FOR I=1 TO N
110 CLS
120 PRINT "ECUACION NUMERO ";I
130 FOR J=1 TO N
140 PRINT "¿CUAL ES EL COEFICIENTE DE X";J;"?";
150 INPUT C(I,J)
160 NEXT J
170 INPUT "¿CUAL ES EL TERMINO INDEPENDIENTE";T(I)
180 NEXT I
190 FOR I=1 TO N
200 FOR J=1 TO N
210 LET D(I,J)=C(I,J)
220 NEXT J:NEXT I
230 IF N>4 THEN PRINT:PRINT "¡UN MOMENTO!, ESTOY HACIENDO LOS CALCULOS"
240 GOSUB 1000
250 LET CRAM=DET
260 CLS
270 FOR A=1 TO N
280 FOR B=1 TO N
290 PRINT C(A,B);"X";B;
300 IF B=N THEN PRINT " =";GOTO 330
310 IF C(A,B+1)<0 THEN PRINT " ";GOTO 330
320 PRINT " +";
330 NEXT B
340 PRINT T(A)
350 NEXT A
360 PRINT
370 IF DET=0 THEN PRINT "EL SISTEMA NO TIENE SOLUCION UNICA.HAY ECUACIONES LINEA
LMENTE DEPENDIENTES" :END
380 IF N>4 THEN PRINT:PRINT "CONTINUO CON MIS CALCULOS":PRINT
390 FOR A=1 TO N
400 FOR B=1 TO N
410 FOR C=1 TO N
420 IF C<>A THEN LET D(B,C)=C(B,C)
430 IF C=A THEN LET D(B,C)=T(B)
440 NEXT C:NEXT B
450 GOSUB 1000
460 PRINT "EL VALOR DE X";A;"ES";DET/CRAM
470 NEXT A
480 END
1000 REM *****
1010 REM **
1020 REM ** SUBROUTINA QUE CALCULA EL DETERMINANTE DE LA MATRIZ **
1030 REM ** D(N,N) **
1040 REM **
1050 REM *****
1060 IF N=1 THEN LET DET=D(1,1):RETURN
1070 LET RANG=N
1080 LET SP=0
1090 GOSUB 2000
1100 RETURN
2000 LET DET=0
2010 IF RANG=2 THEN GOTO 2250
2020 FOR I=1 TO RANG
2030 FOR J=1 TO RANG
2040 LET P(RANG,I,J)=D(I,J)
2050 NEXT J:NEXT I
2060 FOR I=1 TO RANG
2070 FOR J=2 TO RANG
2080 LET COL=1
2090 FOR K=1 TO RANG
2100 IF K<>I THEN LET D(J-1,COL)=P(RANG,J,K):LET COL=COL+1
2110 NEXT K:NEXT J
2120 LET RANG=RANG-1
2130 LET SP=SP+1
2140 LET S(SP)=DET
2150 LET SP=SP+1:LET S(SP)=1
2160 GOSUB 2000
2170 LET I=S(SP):LET SP=SP-1
2180 LET RANG=RANG+1
2190 IF (I MOD 2)=1 THEN LET S(SP)=S(SP)+(P(RANG,1,1)*DET)
2200 IF (I MOD 2)=0 THEN LET S(SP)=S(SP)-(P(RANG,1,1)*DET)
2210 LET DET=S(SP)
2220 LET SP=SP-1
2230 NEXT I
2240 RETURN
2250 LET DET=(D(1,1)*D(2,2))-(D(1,2)*D(2,1))
2260 RETURN
2270 REM Para Commodore cambiar CLS por PRINT CHR$(147)
2280 REM Para Spectrum cambiar END por GOTO 9999 en 370 y 480

```

Este programa utiliza la subrutina 1000, que calcula el determinante de una matriz de dimensión $N \times N$, que se encuentre guardada en la tabla $D(N,N)$. Esta subrutina se trata del programa que ya fue publicado en esta misma sección en el número 23 para realizar dicha operación y que ya se explicó en aquella ocasión.

El programa principal comienza pidiendo los datos del sistema (instrucciones 70-180). En la línea 80 se declaran todas las variables dimensionadas (DIM), que va a utilizar el programa: $C(N,N)$ será la tabla donde se guarden los coeficientes (matriz A), $T(N)$, aquella en la que se almacenen los términos independientes y $D(N,N)$, $P(N,N,N)$ y $S(N^2)$ son tablas que necesita emplear la subrutina y que, como ya se ha dicho antes, su funcionamiento ya fue comentado cuando se explicó ésta.

Entre las instrucciones 190 y 250 se calcula el determinante de la matriz de los coeficientes (recordar que para hallar el determinante de una matriz mediante la subrutina 1000, previamente se debe de haber guardado ésta en la tabla (DIM) D).

Las instrucciones 270-350 son las encargadas de escribir por pantalla el sistema de ecuaciones introducido. El FOR de 390 calculará, cada vez que se repita, el valor de una de las incógnitas, para lo que halla el determinante correspondiente (líneas 400-450) y realiza la división por el que fue calculado anteriormente (línea 460).

Cuando el determinante de la matriz de coeficientes es cero, el sistema no tiene solución única, ya que eso significa que algunas ecuaciones son linealmente dependientes. Esto lo controla el programa en la línea 370.

Notar que para el cálculo de determinantes mayores que cuatro el programa tarda unos segundos en dar la solución aumentando este tiempo de manera progresiva con el número de incógnitas y, por tanto, con la dimensión del determinante. Esto es debido al gran número de operaciones que se tienen que realizar para obtener estos cálculos, como el propio lector sabrá si alguna vez ha tenido que resolver alguno de estos determinantes manualmente. Para los sistemas de más de cuatro incógnitas el programa saca unos mensajes por pantalla en los que indica que está realizando los cálculos para que el lector no se impacienta al ver que no le obtiene los resultados de manera instantánea.

SOCIEDAD

■ Asignación de diputados mediante la aplicación de la regla D'Hont

Vivimos en sistema democrático en el que son comunes las elecciones de representantes en los distintos parlamentos e instituciones. El número de representantes que se selecciona de cada partido o lista que se presenta a las elecciones es proporcional al número de votos que haya obtenido dicho partido en la jurisdicción correspondiente y la regla empleada para esta asignación es la llamada regla D'Hont.

En esta sección se ofrece un programa que realiza la asignación de los diputados correspondientes a una serie de partidos que se han presentado a unas elecciones mediante la aplicación de dicha regla.

Veamos, en primer lugar, el funcionamiento de esta regla D'Hont: en primer lugar, digamos que utiliza tres listas, cuyo tamaño es igual al número de partidos existentes, y que son:

- Una lista donde se tienen los votos obtenidos por cada uno de los partidos.
- Una lista donde se irán poniendo los diputados que vayan correspondiendo a cada uno de los partidos y que antes de comenzar la asignación tendrá, por tanto, un cero en los lugares correspondientes a cada uno de los partidos.
- Una lista auxiliar en la que inicialmente se pondrán los votos obtenidos por cada partido y que posteriormente irá variando, como se indica más adelante.

	Votos		Dip.		Aux.
Partido 1	125.237	P 1	0	P 1	125.237
Partido 2	249.314	P 2	0	P 2	249.314
Partido 3	14.817	P 3	0	P 3	14.817
Partido 4	187.209	P 4	0	P 4	187.209
Partido N	490.281	P N	0	P N	490.281

Situación inicial de las listas antes de comenzar a aplicarse la regla D'Hont.

Figura 5

APRENDER CON EL ORDENADOR

Una vez que se tienen estas tres listas con los valores iniciales indicados, se realizan las siguientes operaciones:

Para cada diputado que haya que asignar se repite:

- Elegir el mayor de la lista auxiliar.
- Asignar el diputado que se está repartiendo al partido elegido anteriormente por ser el que tenía un número mayor en la lista auxiliar.

— Dividir el número de votos del partido elegido en los puntos anteriores entre el número de diputados que tenga asignados ese partido hasta ese momento aumentados en una unidad y guardar el resultado de esta división en el lugar correspondiente a ese partido en la lista auxiliar.

En caso de que en el apartado de seleccionar el mayor de la lista auxiliar haya un empate, la ley regula que se deberá elegir uno de los empatados por sorteo.

Veamos una aplicación de esta regla a un sencillo ejemplo: supongamos que tenemos que repartir tres diputados entre tres partidos que se han presentado a unas elecciones y cuyos votos obtenidos son los siguientes:

Partido 1 : 15.225
Partido 2 : 249.142
Partido 3 : 185.412

Figuras 6

Las tres listas que utiliza la regla D'Hont valdrían inicialmente para este ejemplo:

	Votos	Dip.	Aux.
1	15.225	0	15.225
2	249.142	0	249.142
3	185.412	0	185.412

Figura 7

Como hay que repartir tres diputados, habría que repetir tres veces las operaciones indicadas correspondientes a la regla D'Hont para cada uno de los diputados a asignar. En la siguiente figura vemos cómo irían quedando las tres listas después de repetir estas operaciones cada una de las veces:

	Votos	Dip.	Aux.	
1	15.225	0	15.225	
2	249.142	1	124.571	$249.142/2 = 124.571$
3	185.412	0	185.412	

Asignación del 1.^{er} diputado. El partido 2 era el mayor de la lista Aux.

	Votos	Dip.	Aux.	
1	15.225	0	15.225	
2	249.142	1	124.571	
3	185.412	1	92.706	$185.412/2 = 92.706$

Asignación del 2.^o diputado. El partido 3 era el mayor de la lista Aux.

	Votos	Dip.	Aux.	
1	15.225	0	15.225	
2	249.142	2	83.047,3	$249.142/3 = 83.047,3$
3	185.412	1	92.706	

Asignación del 3.^{er} diputado. El partido 2 era el mayor de la lista Aux.

Figura 8.

En estas elecciones y con este número de votos habrían obtenido, por tanto, dos diputados el partido número 2 y un diputado el número 3 de los tres que había que repartir.

■ Comentario del programa

A continuación vemos el programa que nos realiza la asignación de diputados a una serie de partidos a partir de los votos obtenidos en la jurisdicción correspondiente:

```

10 REM *****
20 REM **
30 REM ** PROGRAMA QUE OBTIENE EL NUMERO DE DIPUTADOS QUE LE CORRESPONDEN **
40 REM ** A UNA SERIE DE PARTIDOS SEGUN LOS VOTOS OBTENIDOS APLICANDO LA **
50 REM ** REGLA D'HONT **
60 REM **
70 REM *****
80 RANDOMIZE TIMER

```

```

85 REM La instruccion 80 solo es necesario ponerla para IBM o compatibles
90 CLS
100 INPUT "CUANTOS DIPUTADOS HAY QUE ASIGNAR EN TOTAL";D
110 CLS
120 INPUT "CUANTOS PARTIDOS SE HAN PRESENTADO A LAS ELECCIONES";P
130 DIM N$(P);DIM VOT(P)
140 FOR I= 1 TO P
150 CLS
160 PRINT "PARTIDO NUMERO ";I
170 PRINT :INPUT "NOMBRE DEL PARTIDO";N$(I)
180 PRINT :INPUT "NUMERO DE VOTOS OBTENIDOS";VOT(I)
190 NEXT I
200 DIM AUX(P);DIM DIP(P);DIM IG(P)
210 FOR I=1 TO P
220 LET AUX(I)=VOT(I)
230 NEXT I
240 FOR I=1 TO D
250 LET MAX=1;LET NIG=1;LET IG(I)=1
260 FOR J=2 TO P
270 IF AUX(J)>AUX(MAX) THEN LET MAX=J;LET NIG=1;LET IG(NIG)=J
280 IF AUX(J)=AUX(MAX) THEN GOTO 310
290 LET NIG=NIG+1
300 LET IG(NIG)=J
310 NEXT J
320 IF NIG<2 THEN GOTO 350
330 LET S=INT(RND(1)*NIG)+1
340 LET MAX=IG(S)
350 LET DIP(MAX)=DIP(MAX)+1
360 LET AUX(MAX)=VOT(MAX)/(DIP(MAX)+1)
370 NEXT I
380 CLS
390 FOR I=1 TO P
400 PRINT "PARTIDO          ";N$(I)
410 PRINT "N. DE VOTOS          ";VOT(I)
420 PRINT "DIPUTADOS OBTENIDOS    ";DIP(I)
430 PRINT :PRINT "PULSA UNA TECLA PARA EL SIGUIENTE"
440 LET A$=INKEY$
450 IF A$="" THEN GOTO 440
460 NEXT I
470 REM Modificaciones para Commodore;
480 REM Cambiar CLS por PRINT CHR$(147)
490 REM 440 GET A$
500 REM Para SPECTRUM; 130 DIM N$(P,30);DIM VOT(P)

```

El programa comienza pidiendo los datos necesarios: número de partidos que se han presentado, número total de diputados que hay que repartir y nombre y votos correspondientes a cada partido.

Las tres listas que se necesitan para la regla D'Hont están guardadas en el programa en las variables dimensionadas (DIM) VOT, DIP y AUX, respectivamente. El programa utiliza además una lista llamada IG, en la que va a ir guardando aquellos partidos con los que se encuentre, al elegir el mayor de la lista AUX, que estén empatados en el primer lugar. La variable NIG controlará el número de partidos que se encuentran empatados.

La aplicación de la regla D'Hont comienza en la línea 240, cuyo FOR se repetirá una vez por cada diputado que haya que asignar. Entre las instrucciones 250 y 310 se selecciona el mayor de la lista AUX, guardándose en MAX el número del partido correspondiente a este mayor, excepto en el caso de que haya varios iguales, que se guardan éstos en la lista IG, como ya se ha indicado anteriormente.

Si se han encontrado varios iguales, en las instrucciones 330 y 340 se selecciona aleatoriamente uno de ellos, que se guarda en MAX.

Las líneas 350 y 360 corresponden a las dos operaciones restantes de la regla D'Hont para el partido que tuviera mayor número, guardado en su lugar correspondiente de la lista AUX.

Las últimas instrucciones del programa son las encargadas de escribir en la pantalla los datos correspondientes a cada partido.

PARA LOS MAS JOVENES

■ Codificación y decodificación de mensajes aplicando el método de las dos ruedas

Una cosa que todos probablemente os hayáis planteado alguna vez es la de tener un sistema para escribir mensajes en algún tipo

APRENDER CON EL ORDENADOR

de código secreto. A continuación os ofrecemos un programa que os va a permitir codificar y decodificar mensajes mediante el llamado método de las ruedas.

Este método consiste en lo siguiente:

En primer lugar, se deben confeccionar dos ruedas que contengan todas las letras del abecedario ordenadas y de tal manera que puedan ser colocadas tal y como se indica en la figura:

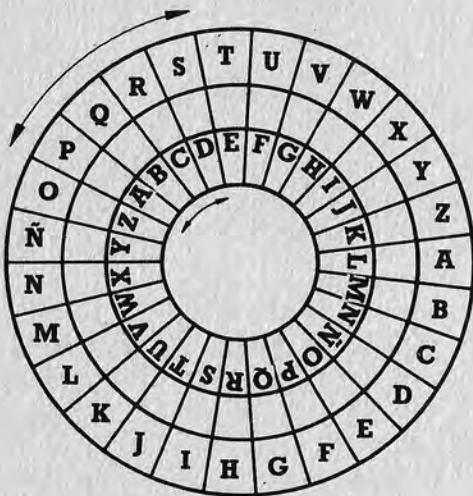


Figura 9

Las ruedas deberán, además, poder girar cada una independientemente de la otra.

Para codificar un mensaje se deben colocar las ruedas de tal forma que la letra A de la rueda exterior coincida con otra letra de la rueda interior. A partir de ahora, lo único que habrá que hacer será buscar con qué letra de la rueda interior se corresponde la que queremos cifrar y poner en el mensaje secreto ésta en vez de la verdadera. Para descifrar el mensaje, lo que habrá que saber será la letra

de la rueda interna con la que se corresponde la A de la rueda externa y, después de colocar las ruedas coincidiendo estas letras, buscar cada letra del mensaje secreto en la rueda interna y ver con cuál de la rueda externa se corresponde.

Supongamos que hemos colocado las ruedas tal y como estaban en la figura anterior y que, por tanto, la letra A de la rueda externa se corresponde con la M de la interna. En este caso, si quisiésemos codificar la palabra TECLA en este código secreto, nos quedaría FPÑWM, ya que las letras de la palabra TECLA se corresponden en la rueda interna con estas otras.

LETRA	CORRESPONDENCIA RUEDA INTERNA
T	F
E	P
C	Ñ
L	W
A	M

Figura 10

Comentario del programa

A continuación vemos el programa que nos realiza la codificación y decodificación de mensajes utilizando este método:

```

10 REM *****
20 REM **
30 REM **   PROGRAMA QUE CODIFICA Y DECODIFICA MENSAJES APLICANDO   **
40 REM **           EL METODO DE LAS DOS RUEDAS                       **
50 REM **
60 REM *****
70 CLS
80 INPUT "que letra se corresponde con la A en la otra rueda";L$
90 IF L$="N" THEN LET DES=14:GOTO 120
100 LET DES=ASC(L$)-ASC("A")
110 IF ASC(L$)>78 THEN LET DES=DES+1
120 CLS
130 PRINT "CODIFICAR UN MENSAJE --> 1"
140 PRINT :PRINT "DECODIFICAR UN MENSAJE-->2"
150 PRINT :INPUT "ELIGE OPCION ",OP
160 CLS
170 PRINT "INTRODUCE A CONTINUACION EL MENSAJE EN MAYUSCULAS:MAXIMO :255 CARACT"
180 INPUT M$
190 IF DES=0 THEN LET N$=M$:GOTO 450
200 FOR I=1 TO LEN(M$)

```

```

210 A$=MID$(M$,I,1)
220 IF OP=2 THEN GOTO 340
230 IF A$="N" THEN LET A=78+DES:GOTO 280
240 IF ASC(A$)<65 OR ASC(A$)>90 THEN LET A=ASC(A$):GOTO 320
250 LET A=ASC(A$)+DES
260 IF A=79 THEN LET A=ASC("N"):GOTO 320
270 IF A>79 AND ASC(A$)<79 THEN LET A=A-1
280 IF A<91 THEN GOTO 320
290 LET A=A-27
300 IF A=78 THEN LET A=ASC("N")
310 IF A<78 THEN LET A=A+1
320 LET N$=N$+CHR$(A)
330 GOTO 440
340 IF A$="N" THEN LET A=79-DES:GOTO 390
350 IF ASC(A$)<65 OR ASC(A$)>90 THEN LET A=ASC(A$):GOTO 430
360 LET A=ASC(A$)-DES
370 IF A=78 THEN LET A=ASC("N"):GOTO 430
380 IF A<78 AND ASC(A$)>78 THEN LET A=A+1
390 IF A>64 THEN GOTO 430
400 LET A=A-27
410 IF A=79 THEN LET A=ASC("N"):GOTO 430
420 IF A>79 THEN LET A=A-1
430 LET N$=N$+CHR$(A)
440 NEXT I
450 PRINT N$
460 REM Para Commodore cambiar CLS por PRINT CHR$(147)
470 REM Para SPECTRUM :210 LET A$=M$(I TO 1)

```

El programa utiliza los códigos internos que tiene el ordenador para cada letra (códigos ASCII), ya que éstos son una numeración por orden alfabético, que va desde el 65 (A) hasta el 90 (Z). El único problema que se plantea el programa para los mensajes en español es la letra Ñ, cuyo código interno no se corresponde con el que debiera entre la N y la O, debido a que este orden alfabético está realizando conforme al abecedario inglés, en el que no existe la Ñ. Esta letra normalmente tendrá un código fuera de los rangos 65-90. Así, el código ASCII de la letra N es el 78 y el de la letra O el 79. El programa en todas sus operaciones tiene que controlar que la Ñ está entre medias. El resto de las letras están ordenadas alfabéticamente internamente en el ordenador con los códigos expresados, excepto las letras compuestas, que deben ser tratadas como dos letras simples.

La técnica fundamental que utiliza el programa para codificar un mensaje es la siguiente: después de pedir la letra que se corresponde con la A en la rueda interna (línea 80), calcula el número de letras que hay entre esa letra y la A, ya que ese número de letras será el que haya que sumar a cada una de las letras que queramos transformar. Para el ejemplo que teníamos anteriormente, entre la letra A y la M (que es en la que se transformaba la A) hay doce letras, por lo que para transformar la letra B habrá que poner la que está doce letras después (la N, siguiente de la M), la C se transformará en la siguiente, y así sucesivamente, teniendo en cuenta que cuan-

do se llegue a la Z hay que continuar contando otra vez por la A. De esta forma, lo único que tiene que hacer el programa es calcular el código de la letra que quiere codificar, sumar a este código el número de letras que hay que desplazar y hallar la letra correspondiente al nuevo código obtenido, teniendo en cuenta el caso comentado antes de la Ñ y cuando se sobrepasa la Z, que hay que volver a empezar.

Veamos cómo realiza esto el programa:

— **Líneas 90-110:** Calcula el número de letras que habrá que sumar a cada letra posteriormente. Este número lo guarda en DES, tratando, en primer lugar, la Ñ, a continuación calculando dicho número como la diferencia entre el código de la A y el de la letra introducida, y teniendo en cuenta después que si la letra está después de la Ñ, este número habrá que aumentarlo en uno, porque hay una letra más (la Ñ) que el programa no ha tenido en cuenta al hallar la diferencia.

Hasta la línea 190 se introducen los datos y el mensaje, que se guarda en M\$.

El FOR de 200 va a ir recorriendo cada una de las letras del mensaje y codificándolas o decodificándolas, según se haya introducido la opción. Las líneas 230-320 se ejecutan en el caso de que se esté codificando y las líneas 340-430 si se está decodificando.

Veamos cómo se realiza la codificación de una letra:

— **Línea 230:** Si es Ñ, se suman las letras que hay que desplazar a 78, y se salta a 280.

— **Línea 240:** Si es un carácter distinto de una letra, se deja como está, ya que estos caracteres especiales como guiones o de otro tipo no hay que codificarlos.

— **Línea 250:** Se suma al código de la letra el número DES.

— **Líneas 260 y 270:** Se tiene en cuenta si al sumar se ha obtenido la Ñ o se ha pasado por ella, teniendo en cuenta que en este último caso el código correcto será uno anterior al obtenido, ya que hay entre medias una letra más que el programa al sumar no ha tenido en cuenta.

— **Líneas 290-310:** Estas líneas se ejecutan cuando al sumar se ha sobrepasado el código de la Z, con lo que hay que volver a co-

menzar desde la A de nuevo. Esto se hace en el programa restando al código obtenido el número total de letras del alfabeto (27), teniendo en cuenta, como siempre, en caso de que se obtenga la Ñ o se haya pasado por ella (sin tenerla en cuenta, evidentemente).

En la **línea 320** se va guardando en N\$ el mensaje ya codificado. Notar que para cada letra que se codifica, el código de la letra que estamos calculando se guarda siempre en la variable A.

El proceso de decodificación de una letra (líneas 340-430) puede observarse que se trata de las mismas operaciones de la codificación, pero realizadas evidentemente al revés.

VOCABULARIO DE INFORMATICA

Reubicar. Término inglés que en Informática se refiere a la operación de mover una rutina de una zona de almacenamiento a otra, y realizar los ajustes de direccionamiento necesarios para que dicha rutina pueda ejecutarse desde su nueva dirección.

Remota, estación. Terminal de comunicaciones que se encuentra distante en tiempo, espacio o eléctricamente del equipo de proceso de datos con el que va a comunicarse.

Remoto, acceso. En comunicaciones, facilidad de proceso de los datos desde una o más estaciones que están distantes del punto en que se encuentra la información.

Repertorio de instrucciones. Conjunto de instrucciones disponibles de un modo inmediato en un microordenador.

Representación numérica. Sistema utilizado para simbolizar los valores numéricos. Es sinónimo de notación numérica. La representación decimal utiliza los diez dígitos decimales (1...9 y el 0), la binaria sólo el 1 y el 0, la exponencial utiliza la mantisa y el exponente, etcétera.

Reset (reinicialización). Devolver un dispositivo a su estado inicial nada más ser conectado a la alimentación o al producirse una condición concreta preestablecida, de ruptura del proceso. Existen dos tipos de reinicialización, una en frío, que borra todo lo que se encontraba en la memoria, y otra en caliente, que envía una señal al microprocesador, permitiendo al operador retomar el control.

Residual, tasa de error. Tasa de error, que subsiste después de realizar una corrección.

Resource (ver Recurso).

Restart (rearranque). Término inglés que indica el restablecimiento de la ejecución de una rutina, utilizando los datos grabados en un determinado punto.

Retrieve (recuperación). Término inglés que designa la acción de recuperar datos o información de un archivo o banco de datos.

Retroacción. Realimentación negativa. Acción por la que se lleva la salida de un sistema a la entrada del mismo, para intentar cambiar su actuación general. Existen dos tipos de realimentación, una positiva y otra negativa. La realimentación negativa o retroacción envía la señal de la salida a la entrada, es decir, se resta de la señal de entrada la salida. (La realimentación positiva consiste en enviar la señal de salida a la entrada, pero "en fase", es decir, sumándola a ésta.)

Return code. Término inglés que indica un código que se utiliza para activar la ejecución de programas, uno tras otro. Es normal que exista un registro especial para almacenar estos códigos.

Rewind. Término inglés que significa rebobinar. Suele referirse a cintas magnéticas o cassettes que se utilizan como sistema de almacenamiento. En algunos lenguajes existe una instrucción *rewind* para ordenar esta acción.

Right justify. Término inglés que en Informática indica el ajuste o alineamiento del texto, por la derecha. También se puede utilizar la justificación izquierda o alineamiento del texto por la izquierda.

Robótica. Ciencia que se ocupa del diseño y aplicaciones para los robots. Estudia los procesos de producción, y las posibilidades de utilizar robots en su mecanización. La robótica está generalmente unida a la informática, ya que, aunque los primeros robots eran autómatas, totalmente mecánicos, los modernos requieren la utilización de ordenadores especializados.

Robotics (véase Robótica).

ROM. Read Only Memory-Memoria sólo de lectura. Dispositivo de almacenamiento de la información, que se caracteriza porque la

información que está almacenada en ella no puede modificarse, aunque sí leerse. Estas memorias se graban en la fábrica, y ya no pueden modificarse. Suelen estar fabricadas utilizando tecnologías avanzadas de integración a gran escala. En los pequeños ordenadores personales es frecuente que el sistema operativo y el intérprete del BASIC se encuentren en ROM. Existen unos tipos especiales de memorias ROM, las PROM, que pueden ser programadas por el usuario, pero sólo la primera vez, y otras, llamadas EPROM que pueden programarse numerosas veces (previo borrado), pero siguiendo sistemas complicados sólo al alcance de los profesionales.

Roundin off. Término inglés que indica el redondeo. (Véase, pues, **Redondeo.**)

Row. Término inglés que indica línea. Se utiliza para multitud de indicaciones, en las aplicaciones junto con la columna (column).

RPG. Report Program Generator-Generador de programas para informes. Lenguaje de programación que se utiliza para la creación y diseño de formularios, impresos, y otras "herramientas" para la creación de documentos.

RS232C. Interface serie utilizado principalmente para conectar impresoras y otros dispositivos de comunicación, relativamente lentos. Velocidades usuales con este interface son las que van desde 110 a 9600 baudios.

Rutina. Parte de un programa, que puede considerarse como otro pequeño programa, es decir, es autónoma.

Salida. Suele indicar el conjunto de datos que se obtienen como resultado de alguna operación, o de la ejecución de un programa o rutina. En el contexto de los elementos hardware se refiere a los dispositivos que pueden tomar o recibir los datos de la UCP para, tras su almacenamiento y preelaboración, presentarlos al usuario.

Salto. Derivación de un punto a otro dentro de un programa o de una rutina. Los programas son ejecutados por el ordenador instrucción por instrucción, de forma secuencial. Sin embargo, en ciertas instrucciones de salto se le da la orden de pasar a otro punto del programa, para seguir de

nuevo secuencialmente las instrucciones. Dentro de estas instrucciones de salto, existen dos tipos: unas ordenan el salto, sean cuales sean las condiciones (salto incondicional); otras, especifican las condiciones que deben cumplirse para que se realice el salto.

Salvar. Transferir, a una memoria permanente datos o programas que se encuentran en la memoria central del ordenador. Estas memorias permanentes suelen ser discos duros, diskettes o cintas magnéticas.

Sampling. Término inglés que significa muestreo. Consiste en tomar valores de una función o de una variable a intervalos de tiempo especificados. En estadística, tomar valores muestra de un conjunto de valores.

Scale factor. Factor de escala. Factor multiplicador con el que deben realizarse los productos de los datos base para que los resultados obtenidos entren dentro de una determinada gama de valores.

Scratch pad. Memoria de trabajo. Memoria auxiliar o buffer. Zona de la memoria en la que los datos se almacenen de forma temporal.

Screen. En inglés, la pantalla de presentación de datos. También puede referirse al conjunto de información que cabe en una pantalla de presentación de los datos.

Scroll. Término inglés que en Informática se refiere al desplazamiento de la información por la pantalla. Este desplazamiento puede también realizarlo el usuario, utilizando para ello unas teclas especiales de las que suelen disponer los ordenadores.

Search. (Véase **Búsqueda.**)

Sector. Los discos tienen pistas (segmentos circulares concéntricos), y éstas, a su vez, están divididas en sectores. Dicha división puede realizarse por hardware, pero también por software. A esta operación se le denomina "sectorización". El contenido de estos sectores se considera siempre de forma unitaria. (Véase formateado...)

Sectorización. División de las pistas de un disco en sectores. Se puede realizar de forma física, por hardware, mediante procedimientos mecánicos, ópticos, etc., o median-

te software, utilizando para ello los códigos adecuados.

Secuencia. Conjunto de elementos ordenados, según un determinado criterio.

Secuencial. Estructuración en la que cada elemento sigue al anterior, y, por tanto, para acceder a uno de ellos es preciso pasar previamente por los anteriores o posteriores. El acceso a los datos puede ser básicamente secuencial, o (en contraposición) directo, aunque hay otras formas híbridas de acceso.

Segmento. Parte de un programa. Un programa complejo y largo suele dividirse en segmentos, cuando no cabe en la memoria del ordenador su totalidad.

Semántica. Ciencia que estudia las relaciones entre los símbolos y su significado...

Semidúplex. (Ver **Dúplex**.)

Semibyte. Medio byte, es decir, cuatro bits consecutivos de dicho byte. A veces se designa como "nibble".

Semiconductor. Material intermedio entre los conductores y los aislantes. Sus cualidades como conductores han sido estudiadas hace pocos años. Los semiconductores pueden transferir carga eléctrica negativa o positiva (huecos). Los semiconductores son la materia prima para la fabricación de los transistores. Los primeros fueron de germanio, pero las últimas tecnologías de miniaturización utilizan prácticamente siempre silicio.

Señal. Onda portadora de información. Puede ser magnética, eléctrica, de presión, electromagnética, etcétera.

Separador. Delimitador. Carácter que se utiliza para delimitar sin ningún tipo de duda elementos en un programa o rutina.

Serie, acceso en. Transmisión de los datos desde o hacia el dispositivo de almacenamiento de forma consecutiva, es decir, uno tras otro. Proceso por el cual los datos se obtienen o colocan en el dispositivo de almacenamiento de tal forma que el tiempo necesario para ello depende directamente de la situación del dato obtenido más recientemente. En contraposición a acceso aleatorio.

Serie, ordenador. En contraposición a ordenador paralelo. Ordenador cuya arquitectura hace que manipule los datos bit a bit, consecutivamente. Los ordenadores paralelo, por el contrario, solapan ciertas actividades, obteniéndose con ellos unos tiempos de proceso mucho menores.

Seie, transmisión en. Transmisión de elementos de señal a intervalos sucesivos. Estos elementos pueden ser transmitidos con o sin interrupciones, pero en ningún caso simultáneamente. En contraposición a transmisión en paralelo.

Serie, puerto. Dispositivo de acceso en el que los datos entran o salen en serie, por un único canal de comunicación.

Símbolo. Elemento al que se da un significado determinado al que debe representar.

Simplex. En contraposición a duplex. Sistema de comunicación que permite transmitir sólo en una única dirección.

Síncrono. Adjetivo que indica dos acontecimientos que están ocurriendo a un mismo tiempo, de forma simultánea tal que pueda establecerse una relación entre ellos, con la precisión requerida. El dispositivo "reloj" del ordenador es el que se ocupa de los sincronismos.

Sistema. En Informática es un conjunto de elementos de hardware y software que constituyen una unidad para el proceso de datos. Sistema operativo, por el contrario, es un conjunto de programas y rutinas que controlan el funcionamiento del ordenador.

Slave. Término inglés que significa "esclavo". Se refiere a un dispositivo controlado por otro, llamado dispositivo principal, o "maestro".

SLSI. Siglas de Super Large Scale Integration. Integración a Muy Gran Escala. Tecnología de fabricación de componentes electrónicos que los miniaturiza hasta poder colocar más de 100.000 transistores en una misma oblea o chip.

Software. Conjunto de programas. Al ser cargados en el ordenador harán que éste realice tareas determinadas. Es un término muy general, ya que en él se incluye el sistema operativo de control del ordenador, junto con otros programas que realizan ta-

VOCABULARIO DE INFORMATICA

reas específicas: software de base, aplicaciones de usuario, etcétera.

Software de base. (Véase **Software del sistema.**)

Software del sistema. Conjunto de programas fundamentales para el funcionamiento del ordenador.

Soporte. Medio físico en el que se graban los datos para su almacenamiento. Pueden ser, por ejemplo, diskettes, discos duros, etcétera.

Sort. Término inglés que significa ordenar, clasificar.

Source/destination. En español, fuente/destino. Términos que se utilizan para indicar la procedencia y el destino de los datos. Un ejemplo muy claro es la copia de datos, en los que es necesario indicar la procedencia de estos y donde (el destino) van a ser copiados.

SPOOL. Siglas de Simultaneous Peropheral Operation On Line. Técnica consistente en mejorar los tiempos de proceso mediante el almacenamiento de la salida dirigida a un periférico lento (como puede ser una impresora) en un periférico más rápido (disco o memoria) para liberar a la CPU. Posteriormente, las rutinas previstas al efecto en el sistema de SPOOL van "liberando" hacia el periférico lento las informaciones almacenadas.

Spreadsheet. Hoja electrónica de cálculo. Programa que muestra en pantalla una cuadrícula susceptible de ser modificada, de forma que una vez introducidas en ella las cifras adecuadas se recalcula automáticamente mostrando siempre los datos actualizados. Es un sistema muy útil para simplificar las tareas de cálculo rutinarias.

SSI. Siglas de Small Scale Integration. Tecnología de fabricación de componentes electrónicos de hace algunos años que permitía integrar dentro de una misma oblea o chip unos diez transistores.

Stack. Término inglés que significa pila (de almacenamiento). En Informática se refiere a unos registros o posiciones de memoria organizados según la técnica LIFO (Last In-First Out. Último en llegar, primero en salir).

Stack pointer. Indicador de la pila. Indicador del último elemento introducido en la pila, que será el primero en salir.

Statement. (Ver **Instrucción.**)

Status register. (Ver **Registro de estado.**)

Storage. Término inglés que significa almacenamiento. En informática se refiere a la memoria, sea cual sea su tipo.

String. (Ver **Cadena.**)

String variables. (Véase **Variables de cadena.**)

Subíndice. Símbolo auxiliar que se añade a otro principal en su parte inferior, para cualificarlo. Las variables de una matriz llevan subíndices que indican la posición de cada elemento.

Subrutina. Programa que forma parte de otro más general, llamado principal, del que depende para su ejecución. Estas rutinas son llamadas por el programa principal, y se pueden ejecutar todas las veces que sea necesario. El programa principal debe "pasar a la subrutina" determinados parámetros, o datos necesarios para la adecuada presentación de ésta. En muchos lenguajes de programación, las subrutinas a su vez, pueden llamar a otras subrutinas según el mismo principio. Esta es la llamada "recursividad".

Supervisor. Sinónimo de sistema operativo. Programa que controla el funcionamiento del ordenador.

Supresión de ceros. Se refiere a la eliminación de los ceros que aparecen a la izquierda de una cifra para que, al presentarla, resulte más clara.

Switch. Término inglés que significa conmutador, interruptor. En Informática se refiere a interruptores o conmutadores físicos, o de hardware, y también a otros realizados con el software.

Sintaxis. Conjunto de reglas para construir las frases de un lenguaje. En Informática es válida esta misma definición. Cada lenguaje tiene unas normas que deben ser seguidas a la hora de programar, para obtener los resultados apetecidos.

Tabla. Conjunto de datos en el que cada elemento está identificado por una etiqueta,

por su posición relativa respecto de los restantes elementos u otra clave cualquiera. Otra acepción distinta de tabla puede ser vector o matriz de elementos.

Tambor magnético. Memoria de masa constituida por un tambor recubierto por una sustancia magnetizable y sobre la que se graba mediante unas cabezas fijas. Las unidades de discos, por el contrario utilizan cabezas móviles, y pueden almacenar mucha menos información.

Tape. Término inglés que significa cinta. Actualmente se refiere casi únicamente a las cintas magnéticas, aunque también puede indicar cintas de papel, etcétera.

Tarea. Cometido específico y concreto. Los trabajos que debe realizar un ordenador se subdividen en "tareas".

Target language. Término inglés que designa un lenguaje objeto, o lenguaje en el que se traducen las instrucciones.

Target program. Programa objeto. Véase **Target language**.

Tarjeta de circuito impreso. Placa especial sobre la que se realiza un circuito electrónico, utilizando técnicas por las que los conductores y aislantes son la propia placa, sobre la que se montan los componentes.

Tarjeta magnética. Pequeña tarjeta (a veces se denomina ficha, por sus dimensiones) sobre la que se graba información por medios magnéticos.

Tarjeta perforada. Cartulina sobre la que se perforaban los datos utilizando un código de perforación. El más utilizado es el de Hollerith.

Tasa. Velocidad. Suele referirse a la tasa o velocidad de transferencia de datos. También puede referirse a la tasa de un código, como medida de la eficacia de dicho código. A mayor redundancia en dicho código, menor tasa.

Tecla. Cada uno de los elementos del teclado. Véase **Teclado**.

Teclado. Conjunto de teclas. Las unidades de trabajo de los ordenadores actuales (así como los ordenadores individuales monopuesto) disponen siempre de un teclado

como uno de los elementos de entrada de los datos. El teclado está formado por teclas alfabéticas ordenadas según el sistema QWERTY, más otras teclas numéricas, generalmente ordenadas como en las calculadoras convencionales, y otras teclas de control, del cursor, etcétera.

Telecomunicación. Transmisión de señales a través de distancias grandes. Un ejemplo puede ser el telégrafo, la televisión, radio, etcétera.

Telemática. Contracción de Telecomunicación e Informática. Ciencia que se ocupa de la Informática aplicada a las Comunicaciones y viceversa. En los primeros tiempos de la Informática, existía el problema de que los ordenadores eran digitales, mientras que las señales emitidas y recibidas eran analógicas. Resuelto éste y otros problemas, hoy en día existen innumerables aplicaciones informáticas en el campo de las comunicaciones, entre las que citaremos el teletexto, las consultas a bases de datos públicas, los coloquios a distancia, etcétera.

Teleproceso. Proceso de datos a distancia. Técnica por la que mediante unos terminales situados lejos de la unidad central, se puede suministrar datos y recibir resultados a distancia.

Teletexto. Técnica por la cual un usuario de televisión puede abonarse para recibir información de un banco de datos a través de uno de los canales de su televisor. El teletexto puede ofrecer todo tipo de información, desde horario de trenes, información sobre el estado de las carreteras, algún tiempo de clases especiales, etcétera. Sin embargo, no es interactivo: el usuario sólo "lee" la información que se le ofrece, pero no puede "responder". Existen otros servicios de utilización de bases de datos incluso interactivos. Es frecuente que se utilicen las líneas telefónicas, ya que en muchos casos la calidad de recepción de la información no es tan importante frente al precio, mucho más económico.

Teletipo. Dispositivo para transmitir mensajes a distancia. Su aspecto es muy semejante a una máquina de escribir. Su operación, funcionamiento y utilización son semejantes a los del teléfono, pero en vez de transmitir palabras, envía textos.

VOCABULARIO DE INFORMATICA

Télex. Red telegráfica internacional para teletipos. A veces al propio teletipo conectado a la red télex se le denomina télex. También se da este nombre al mensaje enviado o recibido en un teletipo a través de la red Télex.

Temporal, almacenamiento. Posiciones de memoria reservadas para el almacenamiento de resultados intermedios. Sinónimo de memoria de trabajo, memoria intermedia, o buffer.

Terminal. Dispositivo de un sistema informático o red de comunicaciones que permite la salida y entrada de los datos.

Terminal gráfico. Terminal diseñado específicamente para la presentación de gráficos. Dispone de algunas características especiales adaptadas a la mejor presentación y fácil creación de gráficos y diagramas, que los terminales convencionales no poseen.

Texto. Secuencia de caracteres que se considera como una entidad si va precedida y terminada por determinados caracteres de control. En los mensajes transmitidos por una red se suele distinguir entre la cabecera o heading (direccionamiento del mensaje, longitud, etcétera) y el texto propiamente dicho.

Threshold. Umbral. Valor a partir del cual se considera que se ha producido una situación o condición distinta de las anteriores.

Truth table. Ver **Tabla de Verdad**.

Tiempo compartido. Utilización de un mismo sistema o dispositivo para dos actividades distintas que se intercalan en la utilización del equipo o "facilidad". Mediante un sistema operativo adaptado a estas necesidades, cada usuario puede utilizar todos los dispositivos (sólo parcialmente, intercalando su uso con otros usuarios), de tal modo que tiene la "sensación" de disponer de todo el equipo para él solo (aunque quizá, los procesos van más lentos).

Tiempo de acceso. Tiempo necesario para leer (tomar) un dato de la memoria o para "escribirlo" (grabarlo) en ésta.

Tiempo de búsqueda. Tiempo necesario para realizar una operación de búsqueda o localización de algún elemento en la memoria o en un dispositivo periférico.

Tiempo de latencia. Tiempo necesario para localizar un determinado sector de una pista de un disco. Este tiempo suele indicarse como un valor medio.

Tiempo real. Se dice que un ordenador trabaja en tiempo real, cuando es capaz de procesar la información que se va produciendo de tal modo que absorbe el conjunto de datos que se generan para ser procesados, sin que se origine ningún retraso y sin que haya almacenamiento de dichos datos. El funcionamiento en tiempo real es una forma de adaptación del ordenador a los fenómenos físicos que se van produciendo y sobre los que hay que interactuar.

Tierra. Tensión de referencia que se hace igual a la del suelo terrestre. Es muy importante que los dispositivos eléctricos estén conectados a tierra, sobre todo cuando se trata de periféricos que emiten-reciben señales.

Tool. Término inglés que significa herramienta. En Informática suele referirse a un programa específico que desarrolla alguna acción repetitiva concreta, como por ejemplo, copiar discos.

Top down (de arriba a abajo). En contraposición a bottom up. Término inglés que indica un criterio de abordar la elaboración de un programa según el cual se parte de consideraciones más generales, para pasar cada vez a definir bloques o situaciones más concretas.

Trabajo. En Informática se refiere a la división de un programa en actividades parciales o "jobs" que deberá realizar el ordenador.

Trace (seguir la pista). Técnica de localización de los errores de un programa. Es un método para seguir paso a paso la ejecución del programa. Va visualizando el número de la línea de programa nada más ejecutarse junto con el estado de los distintos registros.

Track. Término inglés que significa pista (se refiere a las pistas de los discos magnéticos).

Transductor. Dispositivo que convierte magnitudes físicas en otras manteniendo siempre una proporcionalidad entre ellas. Por

ejemplo, puede traducir fuerzas, presiones, etc., a magnitudes eléctricas, mucho más manejables y susceptibles de ser procesadas, para de ellas obtener información (que seguirá el proceso inverso, una vez procesada, mediante actuadores físicos).

Tratamiento por lotes. También se llama proceso batch. Proceso de la información sin intervención del usuario. La información se encuentra en estado de "espera" hasta el momento adecuado para su ejecución.

Tre. Término inglés que significa árbol. Organización de los datos en forma jerárquica.

Truncamiento. Supresión de las cifras menos significativas de un número. Es importante no confundir el truncamiento con el redondeo. Mientras el primero jamás altera la última cifra que permanece (sólo elimina las deseadas), el redondeo sí la altera (ajustándola a la cifra superior o inferior, según se haga el redondeo por exceso o defecto).

TTL. Siglas de Transistor to Transistor Logic, Lógica Transistor a Transistor. Tecnología utilizada para dispositivos rápidos de integración MSI y SSI.

UART. Siglas de Universal Asynchronous Receiver Transmitter, Transmisor Receptor asíncrono universal. Dispositivo que permite conectar de modo asíncrono dispositivos que envían o reciben señales en paralelo con otros que realizan la misma operación en serie.

Unaria, operación. Operación monádica, con un solo operando.

Unblocked. Término inglés que significa no bloqueado. En Informática se refiere a varios registros que no están unidos formando un bloque.

Unidad. Dispositivo que tiene una función especial bien definida. También puede ser un elemento básico. Es un término muy general, ya que puede referirse a muchos tipos de unidad, como unidad lógica, aritmética, de control, central de proceso, etc.

Unidad de disco duro o rígido. Periférico para el almacenamiento de la información sobre todo en los ordenadores personales. Utiliza discos rígidos (usualmente fijos) y puede almacenar mucha más información que los discos flexibles o floppys.

Unidad de disco flexible. Sirve para el almacenamiento de la información en discos flexibles. Dispone de una parte mecánica para el giro de los discos, otra para el posicionamiento de las cabezas y una parte electromagnética para la grabación y obtención de los datos (además de los elementos auxiliares: lógica general de proceso, alimentación, etc.).

Unlock. Término inglés que significa desbloquear, hacer disponible.

USART. Siglas de Universal Synchronous Asynchronous Receiver Transmitter, Transmisor Receptor síncrono asíncrono universal. Dispositivo para establecer conexiones síncronas y asíncronas entre otros dispositivos.

Usuario. Aquella persona que está utilizando el ordenador.

Valor. Cantidad. Suele referirse al que toma una determinada variable.

Variable. Símbolo que representa una magnitud que puede tomar un conjunto de valores. En informática existen muchos tipos de variables, que reciben tratamiento diferente.

Variable booleana. Variable que representa magnitudes, valores lógicos, booleanos, y con las que se realizan las operaciones lógicas AND, OR, etc.

Variables con índice. Variables que pueden ser numéricas o de cadena, pero que están estructuradas, es decir los elementos de ellas se diferencian tanto por el símbolo como por el subíndice. Suelen pertenecer a matrices o a vectores (matrices de dos dimensiones). En la mayoría de los lenguajes de programación, estas variables deben «dimensionarse» es decir debe preverse que el ordenador reserve un espacio de memoria para la variable de índice declarada.

Variables de cadena. Variables que representan cadenas de caracteres. Son adecuadas para las operaciones típicas de cadenas, como extracción de partes de la cadena, unión de varias cadenas, etc.

Variables globales. Variables que pertenecen al programa principal y a sus subrutinas. Pueden ser de varios tipos, de cadena,

VOCABULARIO DE INFORMATICA

numéricas, con índice, etc. En contraposición a variable local.

Variable local. Variable sólo válida en un entorno local de programa, subrutina, etc. Pueden ser como las globales de varios tipos, de cadena, numéricas, con índice, etc. En contraposición a variable global.

Variables numéricas. Variables que representan magnitudes numéricas, con las que se efectúan las operaciones aritméticas.

Vector. Matriz de dos dimensiones. Variable que sólo tiene un índice.

Verificar. Determinar si una operación se ha llevado a cabo satisfactoriamente.

Vídeo. Videoterminal. Pantalla de un terminal o estación de trabajo.

Videodisco. Disco óptico.

Video terminal. Terminal de un ordenador que dispone de un teclado para entrada-salida y una pantalla de presentación de los resultados, mensajes, etc.

Videoterminal gráfico. Videoterminal capaz de presentar adecuadamente imágenes gráficas. Las características de la pantalla son las necesarias para cubrir esta necesidad: mayor tamaño (en ocasiones), mayor definición, etc.

Virtual. En contraposición a real. Suele referirse al caso en el que se «simula» una memoria (virtual) mucho mayor que la memoria real de que se dispone, con el fin de aumentar la flexibilidad del manejo de programas y datos, al disponer de un «espacio de memoria» más amplio. Las «partes» de la «memoria virtual» que no caben en un momento dado en la memoria real y no van a ser utilizadas inmediatamente son almacenadas temporalmente en un dispositivo externo.

VLSI. Siglas de Very Large Scale Integration. Tecnología de fabricación de componentes electrónicos que permite introducir de 10.000 a 100.000 transistores en una sola oblea o chip.

Volátil, memoria. Memoria que no permite conservar la información durante tiempo indefinido. La información se pierde si no se «refresca» o reescribe periódicamente, o si se corta la alimentación.

Wait. Término inglés que significa espera. En informática se refiere a los momentos en los que el ordenador está a la espera de algún acontecimiento exterior o de la ejecución de una instrucción.

Waiting list. Término inglés que significa lista de espera. Se refiere a los programas o trabajos a la espera de ser ejecutados, que forman una "cola" de espera.

Winchester. Tecnología de fabricación de unidades de disco rígidas. Suelen ser discos de gran densidad de grabación, es decir, disponen de una capacidad de almacenamiento considerable.

Word. Término inglés que significa palabra.

Word processing. Tratamiento de textos. Aplicación de los ordenadores para la manipulación de textos. Estos se introducen a través del teclado, y se pueden corregir, editar, desdoblarse, fundir, etc., con mucha facilidad por el usuario. Este también podrá cuidar de su presentación muy sencillamente.

Workstation. Término inglés que significa puesto de trabajo. Lugar físico donde se realiza el trabajo con el ordenador o con el terminal.

Wrap around. Término inglés que significa arrollamiento. Suele referirse a dos líneas de ordenador consecutivas que se "arrollan" para formar una sola.

Write. Término inglés que significa escribir. En informática indica la grabación de datos en la memoria. Es la operación inversa a la "lectura" u obtención de datos de la memoria.

XOR. Es la función lógica OR exclusiva. El resultado de aplicar esta función será verdadero si uno de los dos operandos es verdadero, pero no los dos (a diferencia de lo que sucede en el operador OR).

Zerofill. Operación de relleno con caracteres cero.

Zerolevel address. Dirección del nivel cero. Término inglés que indica una dirección inmediata.

Zero suppression. Supresión de ceros. Eliminación de los ceros no significativos de una cifra.

